

Net8

Administrator's Guide

Release 8.1.5

February 1999

Part No. A67440-01

Net8 Administrator's Guide, Release 8.1.5

Part No. A67440-01

Copyright © 1999, Oracle Corporation. All rights reserved.

Primary Author: Deborah Steiner

Contributors: Pierre Baudin, Toby Close, Santanu Datta, Harvey Eneman, Shuvayu Kanjila, Tong Ming Lee, Ethan Malasky, Sergio Mendiola, Ed Miner, Radhika Shah, Chandar Venkataraman, Norman Woo, and Jun Wu.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle disclaims liability for any damages caused by such use of the Programs.

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the Programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

Oracle, SQL*Net and SQL*Plus are registered trademarks. Oracle8i and Net8 are trademarks of Oracle Corporation. All other company or product names mentioned are used for identification purposes only and may be trademarks of their respective owners.

Contents

Send Us Your Comments xv

Preface..... xvii

Part I Net8 Overview and Concepts

1 Introducing Net8

Net8 Overview..... 1-2

Oracle Connectivity Overview..... 1-3

Configuration Overview..... 1-4

Connection Routing Capabilities 1-6

2 Architecture and Concepts

Net8 Operations 2-2

 Connect Operations..... 2-2

 Data Operations 2-4

 Exception Operations..... 2-4

Net8 Architecture 2-5

 Networking Environments 2-5

 Net8 and the Transparent Network Substrate (TNS)..... 2-6

 Stack Communications 2-6

 Stack Communications in a Typical Oracle Networking Environment 2-9

 Server-to-Server Interaction 2-13

 Stack Communications in an IIOP Environment..... 2-14

 Stack Communications in a JDBC Environment..... 2-16

Network Sessions to Multi-Threaded Servers	2-20
Network Sessions to Dedicated Servers	2-23
Bequeathed Network Sessions to Dedicated Servers	2-25
Net8 and the Listener	2-27
Database Instance Registration to Listeners	2-27
Client Connections.....	2-28
Connect-Time Failover.....	2-29
Load Balancing.....	2-30
Oracle Names Architecture	2-31
Overview.....	2-31
Administrative Regions	2-33
Domains	2-33
Oracle Names As Data Repository.....	2-34
Data Stored in an Oracle Names Server	2-35
Organizing and Naming Network Components	2-37
Using Multiple Regions to Decentralize Administrative Responsibilities.....	2-40
Differences Between Versions of Oracle Names	2-42
Understanding Discovery.....	2-45
Oracle Connection Manager Architecture	2-46
Oracle Connection Manager Processes.....	2-46
Connection Concentration.....	2-48
Net8 Access Control	2-49
Multi-Protocol Support	2-49
Connection Pooling	2-50

Part II Net8 Setup and Configuration

3 Planning Your Network

Planning Overview	3-2
Defining Your Network Layout	3-2
Resolving Net Service Names	3-3
Naming Methods	3-3
Host Naming	3-3
Local Naming	3-5
Centralized Naming Using Oracle Names.....	3-5
External Naming	3-6
Choosing a Naming Method.....	3-8

Improving Large Network Performance	3-10
Managing Connection Requests.....	3-10
Connection Pooling.....	3-11
Connection Concentration.....	3-11
Client Load Balancing.....	3-12
Optimizing Data Transfer by Adjusting the Session Data Unit (SDU) Size	3-13
Persistent Buffer Flushing for TCP/IP	3-14
Configuring Listener Queuesize	3-14
Planning Summary.....	3-15

4 Understanding Release 8.1 Installation and Net8 Products

New Products and Features In This Release	4-2
Net8 Release 8.1 Configuration File Changes	4-8
Configuration File Syntax Changes	4-9
TNSNAMES.ORA Changes	4-15
LISTENER.ORA Changes	4-18
CMAN.ORA Changes	4-19
Obsolete Parameters for Release 8.1	4-20
Net8 Installation and Net8 Products	4-21
Net8 Server Installation	4-22
Net8 Client Installation	4-24
Net8 Products Overview	4-25

5 Considerations for Upgrades and Migration

Net8 and SQL*Net Differences.....	5-2
Obsolete SQL*Net Parameters.....	5-4
Client and Database Compatibility Issues	5-5
Oracle8i Client Release 8.1 Connection to Oracle8i Database Release 8.1	5-5
Oracle8i Release 8.0/Oracle7 Client Connection to Oracle8i Database Release 8.1	5-6
Oracle8i Client Release 8.1 Connection to Oracle8i Database Release 8.0/Oracle7 Database ...	5-8
Oracle Names.....	5-10
Using the Net8 Assistant to Handle Compatibility Issues.....	5-11
Using Release 8.1 Features and Connecting To a Release 8.1 Service	5-12
Using Release 8.0/7.x Features and Connecting To a Release 8.0/7.x Service.....	5-14
Migrating and Upgrading Considerations	5-16

Upgrading and Migrating to Net8 Release 8.1	5-17
Step 1: Verify Service Name and Instance Name.....	5-17
Step 2: Perform Software Upgrade or Migration on the Server.....	5-18
Step 3: Perform Software Upgrade or Migration on the Client	5-19
Step 4: Perform Functional Upgrade and Migration.....	5-19
Additional Considerations for Oracle Connection Manager and Oracle Names.....	5-21
Using Oracle Connection Manager Release 8.1.....	5-21
Using Oracle Connection Manager Instead of Oracle Multi-Protocol Interchange.....	5-22
Using Oracle Names Version 8.....	5-23

6 Configuring Naming Methods and the Listener

Configuration Overview.....	6-2
Setting Service Names and Instance Names.....	6-3
Configuration Models	6-4
Naming Methods	6-5
Configuration Files	6-6
Configuring the Network with the Host Naming Method	6-7
Step 1: Configure the Listener with the Global Database Name	6-8
Step 2: Configure HOSTNAME as the First Naming Method	6-9
Step 3: Set Up Host Name Resolution Environment	6-11
Configuring the Network with the Local Naming Method.....	6-12
Configuration During Installation	6-13
Configuration After Installation.....	6-14
Configuring the Network with the Oracle Names Method.....	6-21
Step 1: Consider Oracle Names Options	6-23
Step 2: Install.....	6-24
Step 3: Create an Oracle Names Server	6-25
Step 4: Configure Clients and Database Servers To Use Oracle Names Servers.....	6-37
Step 5: Configure the Listener.....	6-40
Step 6: Register Data with the Oracle Names Server.....	6-41
Step 7: (Optional) Delegate Domains to Delegated Administrative Regions	6-50
Configuring the Network with the External Naming Method.....	6-51
Configure NIS.....	6-51
Configure NDS.....	6-55

Configuring the Listener	6-62
Re-Starting the Listener After a LISTENER.ORA File Modification.....	6-64
Configuring Listener Protocol Addresses.....	6-64
Configuring a Listener that Uses a Non-Default Port.....	6-67
Using a Listener Name Other than LISTENER.....	6-69
Handling Larger Volumes of Connection Requests.....	6-70
Configuring a Listener with Service Information.....	6-71
Configuring Protocol Addresses	6-76
Address Components.....	6-76
Protocol Parameters	6-78

7 Enabling Optional Net8 Features

Modifying Net Service Names	7-2
Modifying Attributes of a Net Service Name.....	7-2
Renaming a Net Service Name.....	7-3
Adding a Protocol Address to a Net Service Name	7-3
Deleting a Protocol Address from a Net Service Name.....	7-5
Deleting a Net Service Name	7-6
Configuring Advanced Connection Attributes for a Net Service Name.....	7-7
Configuring Multiple Address Options	7-11
Configuring a Profile	7-15
Configuring Domains	7-16
Prioritizing Net Service Name Naming Method Resolution	7-18
Configuring Preferred Oracle Names Servers.....	7-20
Configuring Advanced Profile Information	7-23
Routing Connection Requests.....	7-26
Configuring Oracle Advanced Security	7-28
Enabling Connection Concentration	7-31
Step 1: Configure Oracle Connection Manager.....	7-31
Step 2: Configure the Server	7-32
Step 3: Configure the Client	7-33
Enabling Multi-Protocol Support	7-35
Step 1: Configure Oracle Connection Manager.....	7-35
Step 2: Configure the Client	7-36

Enabling Net8 Access Control	7-38
Step 1: Configure Oracle Connection Manager.....	7-38
Step 2: Configure the Client	7-40
Configuring the Client for Oracle Connection Manager Features	7-41
Step 1: Configure an Oracle Connection Manager Address.....	7-43
Step 2: Configure the Listener Address.....	7-46
Enabling Connection Pooling	7-49
Configuring Java Option Connections	7-50
Java Stored Procedure Configuration.....	7-51
Enterprise JavaBeans and CORBA Configuration.....	7-51
Using the Default Configuration of Oracle8i for EJBs and CORBA through the Listener	7-54
Using a Non-Default Configuration	7-56
Using SSL with EJBs and CORBA	7-64
Configuring Non-Oracle Database Services	7-65
Configuring Net8 for External Procedures	7-65
Configuring Net8 for Oracle Heterogeneous Services.....	7-67
Configuring Net8 for an Oracle Rdb Database	7-71

8 Establishing a Connection and Testing the Network

Connecting to a Database	8-2
Net8 Component Startup Overview	8-2
Entering a Connect String.....	8-8
Checklist for Troubleshooting Common Startup and Connection Problems	8-11
Testing the Network	8-14
Using Net8 Control Utilities.....	8-15
Testing an Oracle Names Server	8-16
Testing Network Objects Using NAMESCTL or the Net8 Assistant	8-17
Testing a Listener.....	8-18
Testing Oracle Connection Manager	8-18
Testing Configuration on the Server.....	8-18
Testing Network Connectivity from the Client.....	8-20

9 Configuring Multi-Threaded Server

Overview	9-2
Configuring Oracle for Multi-Threaded Server Architecture	9-4
Setting the Initial Number of Dispatchers	9-8
Configuring Both Multi-Threaded Server and Dedicated Server Modes	9-10
Overriding Multi-Threaded Server Mode on the Client	9-11

10 Enabling Net8 Enhancements for Programmers

Net8 OPEN	9-2
Net8 OPEN API Function Calls	9-3
Finding the Net8 OPEN Applications Program Interface	9-8
Building Your Own Application	9-8
Configuring the System to Use Your Net8 OPEN Application	9-9
Sample Programs	9-11
Net8 OPEN API Errors	9-12
UNIX Client Programming	9-13
Signal Handler and Alarm Programming	9-14
Bequeath.....	9-14

11 Troubleshooting Net8

Diagnosing Net8	11-2
Server Diagnostics	11-3
Client Diagnostics.....	11-4
Resolving the Most Common Error Messages	11-7
Troubleshooting Tips from the Field	11-13
Questions to Ask When Troubleshooting.....	11-14
Troubleshooting Network Problems Using Log and Trace Files	11-15
Logging Error Information	11-15
Error Stacks.....	11-16
Log File Names	11-18
Understanding and Setting Log Parameters	11-19
Setting Log Settings During Runtime of Control Utilities.....	11-24
Using Log Files.....	11-25
Listener's Log Audit Trail	11-25
Understanding Oracle Connection Manager Logs.....	11-27

Tracing Error Information	11-31
Trace File Names.....	11-32
Understanding and Setting Trace Parameters.....	11-32
Setting Trace Settings During Runtime of Control Utilities.....	11-39
Evaluating Net8 Traces	11-40
Using the Trace Assistant to Examine Your Trace Files.....	11-43
Contacting Oracle Customer Support	11-58

Part III Reference

A Control Utilities for the Listener, Oracle Names Server, and Oracle Connection Manager

Operating Modes	A-2
Listener Control Utility (LSNRCTL)	A-3
SET and SHOW Modifiers.....	A-3
Security.....	A-3
LSNRCTL Commands	A-4
Oracle Names Control Utility (NAMESCTL)	A-31
Parameter Options.....	A-31
SET and SHOW Modifiers.....	A-31
NAMESCTL's Distributed Operation.....	A-32
Security.....	A-32
Confirmation Mode in NAMESCTL.....	A-33
NAMESCTL Commands	A-34
Oracle Connection Manager Control Utility (CMCTL)	A-89
CMCTL Commands	A-90

B Sample Configuration Files

Profile (SQLNET.ORA)	B-2
Local Naming Configuration File (TNSNAMES.ORA)	B-5
Listener Configuration File (LISTENER.ORA)	B-6
Oracle Names Server Configuration File (NAMES.ORA)	B-7
Oracle Connection Manager Configuration File (CMAN.ORA)	B-9

C Configuration Parameters

Syntax Rules for Configuration Files	C-2
Further Syntax Rules for Configuration Files.....	C-3
Network Character Set.....	C-4
Character Set	C-5
ADDRESSES and ADDRESS_LISTs	C-5
Profile Parameters (SQLNET.ORA)	C-7
Local Naming Parameters (TNSNAMES.ORA)	C-31
General Syntax	C-31
Multiple Descriptions	C-32
Multiple Address Lists.....	C-33
Changed Functionality of Client Load Balancing and DESCRIPTION_LISTs.....	C-34
Examples	C-36
DESCRIPTIONS and DESCRIPTION_LISTs.....	C-38
ADDRESSES and ADDRESS_LISTs	C-39
Optional Parameters for Lists	C-40
CONNECT_DATA Section	C-43
Listener Parameters (LISTENER.ORA)	C-50
Listener Address Section	C-51
SID_LIST_listener_name Static Service Section	C-57
Control Parameters.....	C-62
Oracle Names Parameters (NAMES.ORA)	C-65
Oracle Connection Manager Parameters (CMAN.ORA)	C-76
Protocol-Specific Parameters (PROTOCOL.ORA)	C-81

Glossary

Index

List of Figures

1-1	Basic Client-to-Server Connection	1-2
1-2	Client-to-Server Connection through Oracle Names	1-5
1-3	Client to Server Connections through Oracle Connection Manager	1-6
2-1	OSI Communications Stack	2-7
2-2	Typical Communications Stack in an Oracle environment.....	2-10
2-3	Typical Client vs. IIOP Client Communication Stacks	2-14
2-4	Client/Server Communications Stack in an IIOP Environment	2-15
2-5	Typical Client vs. JDBC Client Communication Stacks.....	2-17
2-6	Client/Server Communications Stack in a JDBC Environment	2-19
2-7	Redirected Connection to a Dispatcher.....	2-22
2-8	Redirected Connection To a Prespawnd Dedicated Server Process	2-24
2-9	Bequeathed Connection To a Dedicated Server Process	2-26
2-10	Listener In a Typical Net8 Connection.....	2-28
2-11	Oracle Names.....	2-32
2-12	Single Domain Naming Model.....	2-37
2-13	Hierarchical Naming Model	2-38
2-14	Default Domains.....	2-39
2-15	Delegated Administrative Regions.....	2-41
2-16	Connection Concentration through Oracle Connection Manager	2-48
2-17	Multi-Protocol Support Through Oracle Connection Manager	2-50
2-18	Connection Pooling.....	2-51
5-1	Oracle8i Client to Oracle8i Database	5-5
5-2	Oracle8i Client release 8.0/Oracle7 Client to Oracle8i Database	5-6
5-3	Oracle8i Client to Oracle8i Release 8.0/Oracle7 Database.....	5-8
6-1	Net Service Name Usage in TNSNAMES.ORA	6-12
6-2	Multiple Listener Addresses.....	6-13
7-1	TNSNAMES.ORA File with and without Oracle Connection Manager	7-34
7-2	CMAN Parameter Configured to Use Multi-Protocol Support.....	7-37
8-1	Successful Trace Route	8-24
8-2	Trace Route with Error	8-24
9-1	Oracle Multi-Threaded Sever Architecture	9-3
10-1	Net8 OPEN	9-2
11-1	Network Products and Error Stack Component.....	11-16
11-2	Typical Error Stack	11-17
11-3	Typical Audit Trail Information for Successful Reload Request.....	11-25
11-4	Typical Audit Trail Information for Successful Connection Request.....	11-26
11-5	CMAN_PID.LOG	11-27
11-6	CMADM_PID.LOG.....	11-27
11-7	Packet Information	11-41
11-8	Trace File Error Output	11-42

11-9	Typical Trace Assistant Conversion	11-44
11-10	Summary Data Packets Sent in a Bequeathed Connection.....	11-45
11-11	Detailed Data Packets Sent in a Bequeathed Connection.....	11-46
11-12	Summary Data Packets Sent in a Redirected Connection	11-47
11-13	Data Packet	11-47
11-14	Two Task Common Summary Information	11-48
11-15	Two Task Common Summary Information	11-50
11-16	Detailed SQL information on top of summary Two-Task.....	11-51
11-17	Typical Trace File Statistics	11-52
11-18	Trace File Example	11-53
C-1	General Syntax of TNSNAMES.ORA	C-31
C-2	Multiple Descriptions in TNSNAMES.ORA	C-32
C-3	Multiple Address Lists in TNSNAMES.ORA.....	C-33
C-4	Client Load Balancing for Release 8.1 and Prior to Release 8.1	C-35
C-5	Multiple Oracle Connection Manager Addresses TNSNAMES.ORA	C-36
C-6	Client Load Balancing in TNSNAMES.ORA.....	C-37

Send Us Your Comments

Net8 Administrator's Guide, Release 8.1.5

Part No. A67440-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the chapter, section, and page number (if available). You can send comments to the Information Development department in the following ways:

- Electronic mail - infodev@us.oracle.com
- FAX - (650) 506-7228 Attn: Net8
- Postal service:
Oracle Corporation
Server Documentation Manager
500 Oracle Parkway, 4OP12
Redwood Shores, CA 94065
USA

If you would like a reply, please give your name, address, and telephone number below.

If you have problems with the software, please contact your local Oracle World Wide Support Center.

Preface

The *Net8 Administrator's Guide* provides the information you need to understand and use the Net8 product and its related applications.

Note: This guide contains information that describes the Net8 features and functionality within the Oracle8i and the Oracle8i Enterprise Edition products. Oracle8i and the Oracle8i Enterprise Edition have the same basic features. However, several advanced Net8 features are available only with Oracle8i Enterprise Edition. For example, to use Oracle Connection Manager, you must have Oracle8i Enterprise Edition. For information about the differences between Oracle8i and the Oracle8i Enterprise Edition and the features and options that are available to you, please refer to *Getting to Know Oracle8i*.

This Preface includes the following topics:

- [Audience](#)
- [Structure](#)
- [Related Documents](#)
- [Conventions](#)
- [Your Comments Are Welcome](#)

Audience

The information in this guide is intended primarily for network or database administrators (DBAs) responsible for making connection to services, such as the Oracle8i database, and network configuration. This guide is also provided for anyone who wants to understand how Net8 works.

Structure

This guide contains 11 chapters and three appendices

Part I: Net8 Overview and Concepts

Chapter 1, "Introducing Net8"	Introduces the Net8 release 8.1 product.
Chapter 2, "Architecture and Concepts"	Describes the Net8 architecture, Net8 interaction in a multi-threaded server or dedicated server environment, Oracle Names, Oracle Connection Manager, and connection pooling.

Part II: Net8 Setup and Configuration

Chapter 3, "Planning Your Network"	Describes considerations for planning a network using Net8. It explains the relationships of the Net8 products, and options for better managing your future network.
Chapter 4, "Understanding Release 8.1 Installation and Net8 Products"	Describes release 8.1 configuration files and installed Net8 products and applications.
Chapter 5, "Considerations for Upgrades and Migration"	Describes upgrading from Net8 release 8.0 to Net8 release 8.1, and migrating from SQL*Net version 2 to Net8 release 8.1.
Chapter 6, "Configuring Naming Methods and the Listener"	Describes how to configure the network using various Net8 methods.
Chapter 7, "Enabling Optional Net8 Features"	Describes how to configure advanced and optional Net8 features, such as Oracle Connection Manager features, connection pooling, client load balancing, connect-time failover, Heterogeneous services, External Procedures, and Oracle Rdb database.
Chapter 8, "Establishing a Connection and Testing the Network"	Describes how to start Net8 components, establish a connection, and test a connection.
Chapter 9, "Configuring Multi-Threaded Server"	Describes how to tune the components of the multi-threaded server architecture.
Chapter 10, "Enabling Net8 Enhancements for Programmers"	Describes Net8 enhancements for programmers. This includes a review of Net8 OPEN, UNIX signal handling, and bequeath adapter.
Chapter 11, "Troubleshooting Net8"	Describes procedures to troubleshoot Net8. This includes information on tracing and logging.

Part III: Reference

Appendix A, "Control Utilities for the Listener, Oracle Names Server, and Oracle Connection Manager"	Describes all commands for Net8 Control Utilities including Listener Control (LSNRCTL), Oracle Names Control (NAMESCTL), and Connection Manager Control (CMCTL).
Appendix B, "Sample Configuration Files"	Provides sample configuration files for profiles (SQLNET.ORA), local names (TNSNAMES.ORA), listener (LISTENER.ORA), Oracle Names servers (NAMES.ORA), and Oracle Connection Managers (CMAN.ORA).
Appendix C, "Configuration Parameters"	Lists and describes configuration parameters for profiles, local names, listener, Oracle Names servers, Oracle Connection Manager, and protocols.

Related Documents

- *Oracle8i Concepts*
- *Oracle8i Reference*
- *Oracle8i Distributed Database Systems*
- *Oracle8i Tuning*
- *Oracle Enterprise Manager Administrator's Guide*
- *Oracle Advanced Security Administrator's Guide*

Conventions

The following conventions are used in this guide:

Convention	Example	Meaning
UPPERCASE	LISTENER.ORA	Calls attention to SQL commands, keywords, file names, and initialization parameters. Most Oracle files names on UNIX are lowercase.
lowercase courier	<code>\$ORACLE_HOME/network/admin startup;</code>	Indicates directories and commands. Most directories and commands on UNIX are lowercase.
<i>Italic</i>	Italic is used to indicate a variable: <i>filename</i> Italic is used for emphasis: The WHERE clause may be used to <i>join</i> rows in different tables.	Indicates a value that you must provide. For example, if a command asks you to type <i>filename</i> , you must type the actual name of the file. Italic is also used for emphasis in the text and to indicate the titles of other guides.
Square brackets []	<code>change_password [<i>listener name</i>]</code>	Encloses optional items.
Angle brackets < >	<code>names.default_domain = <<i>valid domain name</i>></code>	Enclose user-supplied names.
Choose Start >	Choose Start > Programs > Oracle- <i>HOME_NAME</i> > Oracle Enterprise Management > Enterprise Manager Console	Indicates how to start a program on Windows platforms.
Forward slash (\) before a directory name	<code>/network</code>	A subdirectory on UNIX platforms.
Backslash (\) before a directory name	<code>\network</code>	A subdirectory on Windows platforms.
<code>\$ORACLE_HOME</code> on UNIX and <i>ORACLE_HOME</i> on Windows platforms	Go to the <code>\$ORACLE_HOME\network</code> directory	In this Optimal Flexible Architecture (OFA)-compliant release, all subdirectories are no longer under a top level <code>ORACLE_HOME</code> directory. There can be an <code>ORACLE_BASE</code> directory, which may contain multiple Oracle home directories.

Convention	Example	Meaning
Symbols	period . comma , hyphen - semicolon ; colon : equal sign = backslash \ single quote ' double quote " parentheses ()	Symbols other than brackets and vertical bars must be entered in commands exactly as shown.

Your Comments Are Welcome

We value and appreciate your comments as an Oracle user and reader of the user guides. As we write, revise, and evaluate our documentation, your opinions are the most important input we receive. Please use the reader's comment form to tell us what you like or dislike about this manual or other Oracle manuals. If the form is not available, please send your comments to the Information Development department using any of the following:

- E-mail: infodev@us.oracle.com
- FAX - (650) 506-7228. Attn: Net8
- Postal Service:
 Oracle Corporation
 Server Documentation Manager
 500 Oracle Parkway, 4OP12
 Redwood Shores, CA 94065
 U.S.A.

Part I

Net8 Overview and Concepts

Part I provides an overview of Net8 concepts and terminology. It contains the following chapters:

- [Chapter 1, "Introducing Net8"](#)
- [Chapter 2, "Architecture and Concepts"](#)

Introducing Net8

This chapter introduces Net8, and provides an overview of its main applications, features, and functionality. It contains the following sections:

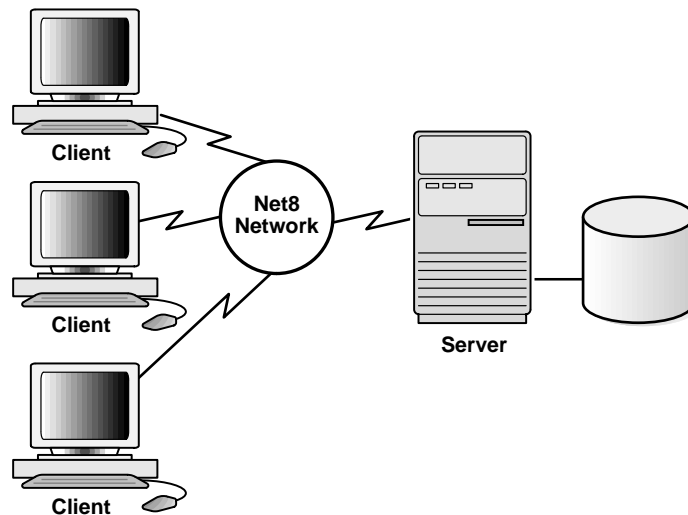
- [Net8 Overview](#)
- [Oracle Connectivity Overview](#)
- [Configuration Overview](#)
- [Connection Routing Capabilities](#)

Net8 Overview

Net8 is the foundation of Oracle's family of networking products, allowing services and their applications to reside on different computers and communicate as peer applications. The main function of Net8 is to establish network sessions and transfer data between a client machine and a server or between two servers. Net8 is located on each machine in the network. Once a network session is established, Net8 acts as a data courier for the client and the server.

Figure 1-1 shows a client-to-server connection:

Figure 1-1 Basic Client-to-Server Connection



Network sessions are established with the help of a listener. The listener is a separate process that resides on the server whose responsibility is to listen for incoming client connection requests and manage the traffic to the server.

The listener brokers the client request, handing off the request to the server. Every time a client (or server acting as a client) requests a network session with a server, a listener receives the actual request. If the client's information matches the listener's information, the listener grants a connection to the server.

For environments where a large number of connections need to access the same service, Net8 offers a routing process called Oracle Connection Manager that resides on a machine separate from the client or server. For further information about Oracle Connection Manager, see "[Connection Routing Capabilities](#)" on page 1-6.

Oracle Connectivity Overview

Net8 allows connections to various services, such as Oracle databases, non-Oracle databases, gateways, and external procedures (functions that can be called from PL/SQL code). As an Oracle database is the most common service in an Oracle environment, this section will focus on database services.

When an end user connects to a database service from across the network, a *connect string* identifies the service through a net service name. For example:

```
sql> CONNECT username/password@net_service_name
```

A *net service name* is able to access a service across the network by providing the network description information necessary to locate the service on the network. A net service name is resolved into the:

- network route to the service, including the listener location through a protocol address
- service name that is typically the *global database name*—a name comprised of the database name and database domain—used to identify the database service

The listener, through a protocol, accepts the client connection. It verifies the client information with the information it has received from the database service, as well as information it has stored in its own configuration file, LISTENER.ORA. If the information matches, a connection is granted.

Configuration Overview

Configuration is accomplished by creating a list of the net service names that map to services and addresses of listener destinations and configuring a naming method by which to resolve the net service name. A *naming method* is a method by which a net service name is resolved.

Wherever you choose to store your service names, Net8 offers tools to easily configure net service names. Net8 supports several categories of naming methods:

- existing name resolution service

Net service names may be stored via an existing IP address translation mechanism.

- client configuration file

Net service names can be stored in a local configured file called TNSNAMES.ORA on each client and server in the network. Because the TNSNAMES.ORA can be configured on individual clients, it allows you to fine tune for a particular client's needs.

- an external naming service

If a third-party naming service is already configured in your environment, Net8 supports storing net service name information in it to avoid additional configuration.

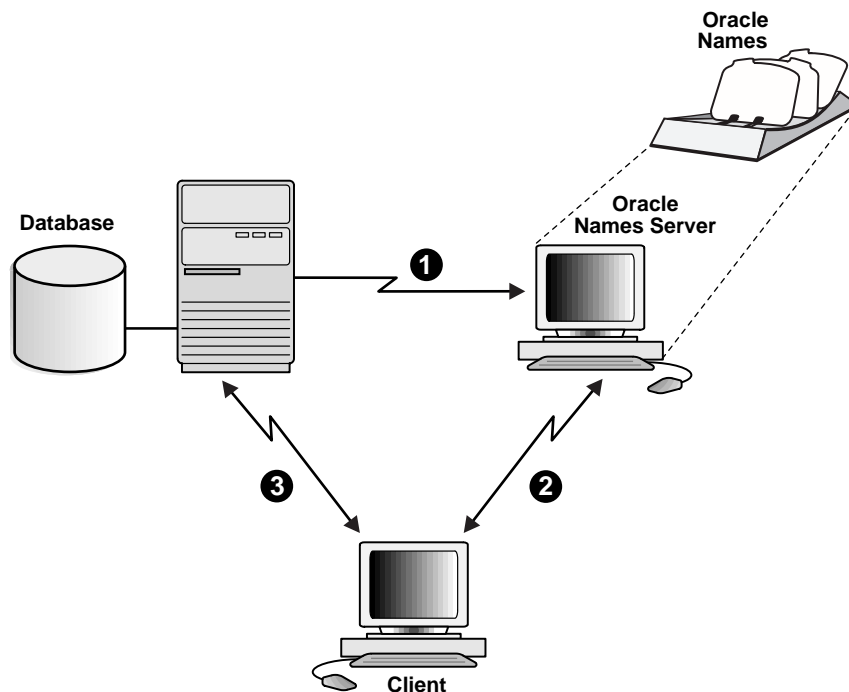
- centralized store

Net service names and service addresses can be stored in a central directory of service addresses. This central store, called Oracle Names, is an Oracle-specific name service that maintains a central store of service addresses. When client applications request connections to servers using a simple name, the Net8 client contacts Oracle Names to resolve the service into a network address descriptor.

Figure 1-2 shows:

1. A database registering its service with an Oracle Names server.
2. Client seeking to locate a service on the network by contacting an Oracle Names server to retrieve the network address.
3. Clients then transparently connecting to the service:

Figure 1-2 Client-to-Server Connection through Oracle Names



For more information about choosing the place to store net service names, see ["Resolving Net Service Names"](#) on page 3-3.

Connection Routing Capabilities

Net8 offers connection routing for environments where:

- a large numbers of users need to access a single service through a single protocol
- different protocols are installed on the client and server, making a typical connection fail

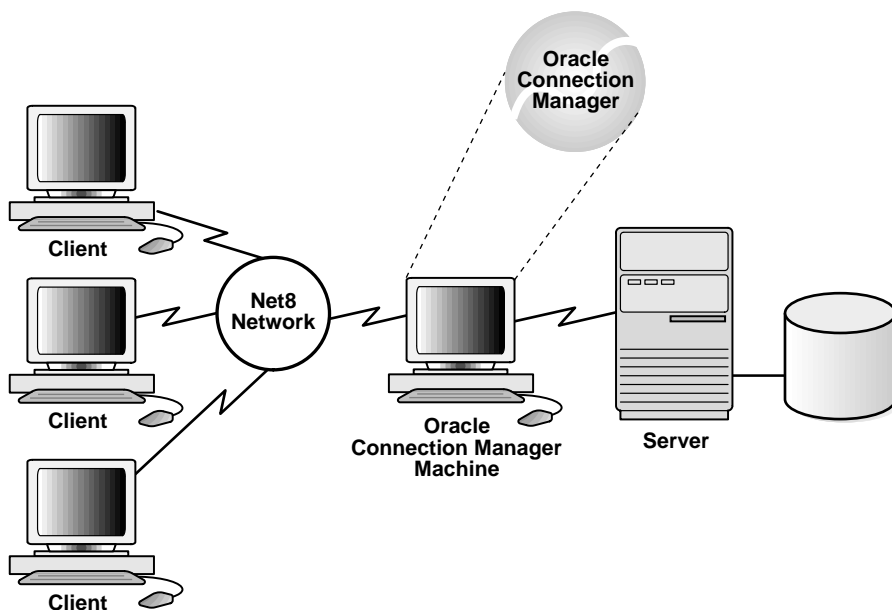
Typical Net8 connections require the client and server to have the same protocol installed.

- control of client access to designated servers in a TCP/IP environment is required

Connection routing is supported by Oracle Connection Manager, through which client connection requests are routed to the server.

Figure 1-3 shows how client connection are routed to Oracle Connection Manager, which resides on a separate machine in the network:

Figure 1-3 Client to Server Connections through Oracle Connection Manager



See "[Oracle Connection Manager Architecture](#)" on page 2-46, for further information.

Architecture and Concepts

Net8 uses the Transparent Network Substrate (TNS) and industry-standard networking protocols to connect a client to a server and establish an Oracle network session.

This chapter describes TNS and the role it plays in distributed systems. It also explains how Net8 interacts with TNS to perform basic connectivity and transport operations. This chapter contains the following sections:

- [Net8 Operations](#)
- [Net8 Architecture](#)
- [Network Sessions to Multi-Threaded Servers](#)
- [Network Sessions to Dedicated Servers](#)
- [Net8 and the Listener](#)
- [Oracle Names Architecture](#)
- [Oracle Connection Manager Architecture](#)
- [Connection Pooling](#)

Net8 Operations

Net8 is responsible for enabling communications between the cooperating partners in an Oracle distributed transaction, whether they be client-server or server-server. Specifically, Net8 provides three basic networking operations:

- [Connect Operations](#)
- [Data Operations](#)
- [Exception Operations](#)

Connect Operations

Net8 supports two types of connect operations:

- [Connecting to Servers](#)
- [Disconnecting from Servers](#)

Connecting to Servers

Users initiate a connect request by passing a user name and password along with an identifier for the service to which they wish to connect. That identifier, called a *net service name*, is mapped to a network address contained in a *connect descriptor*. A connect descriptor contains the:

- network route to the service, including the location of the listener through a protocol address
- service name, a logical representation of a database, used to identify the release 8.1 services, or Oracle System Identifier (SID) used to identify a pre-release 8.1 databases
- optional instance name of the release 8.1 service to which to connect

Clients and servers use this net service name when making a connection with an application.

Depending upon your specific network configuration, a connect descriptor may be stored in one of the following:

- a local names configuration file called TNSNAMES.ORA
- an Oracle Names server for use by Oracle Names
- a external naming service such as NIS or DCE CDS

Net8 coordinates its network sessions with the help of a listener.

Disconnecting from Servers

Requests to disconnect from the server can be initiated in the following ways:

- [User-Initiated Disconnect](#)
- [Additional Connection Request](#)
- [Abnormal Connection Termination](#)
- [Timer Initiated Disconnect or Dead Connection Detection](#)

User-Initiated Disconnect A user can request a disconnection from the server when a client-server transaction completes. A server can also disconnect from a second server when all server-server data transfers have been completed, and no need for the link remains.

Additional Connection Request If a client application is connected to a server and requires access to another user account on the same or on another server, most Oracle tools first disconnect the application from the server to which it is currently connected. Once the disconnection is completed, a connection request to the new user account on the appropriate server is initiated.

Abnormal Connection Termination Other components occasionally disconnect or abort communications without giving notice to Net8. In this event, Net8 recognize the failure during its next data operation, and clean up client and server operations, effectively disconnecting the current operation.

Timer Initiated Disconnect or Dead Connection Detection Dead connection detection allows Net8 to identify connections that have been left hanging by the abnormal termination of a client. This feature minimizes the waste of resources by connections that are no longer valid. It also automatically forces a database rollback of uncommitted transactions and locks held by the user of the broken connection.

On a connection with dead connection detection enabled, a small probe packet is sent from server to client at a user-defined interval (usually several minutes). If the connection is invalid (usually due to the client process or machine being unreachable), the connection is closed when an error is generated by the send operation, and the server process terminates the connection.

Data Operations

Net8 supports four sets of client-server data operations:

- send data synchronously
- receive data synchronously
- send data asynchronously
- receive data asynchronously

On the client side, a SQL dialogue request is forwarded using a send request in Net8. On the server side, Net8 processes a receive request and passes the data to the database. The opposite occurs in the return trip from the server.

Basic send and receive requests are synchronous. When a client initiates a request, it waits for the server to respond with the answer. It can then issue an additional request.

Net8 adds the capability to send and receive data requests asynchronously. This capability was added to support the Oracle shared server, also called a multi-threaded server.

Exception Operations

Net8 supports three types of exception operations:

- initiate a break over the connection
- reset a connection for synchronization after a break
- test the condition of the connection for incoming break

The user controls only one of these three operations, that is, the initiation of a break. When the user presses the Interrupt key (Ctrl-C on some machines), the application calls this function. Additionally, the database can initiate a break to the client if an abnormal operation occurs, such as during an attempt to load a row of invalid data using SQL*Loader.

The other two exception operations are internal to products that use Net8 to resolve network timing issues. Net8 can initiate a test of the communication channel, for example, to see if new data has arrived. The reset function is used to resolve abnormal states, such as getting the connection back in synchronization after a break operation has occurred.

Net8 Architecture

This section covers the following architectural concepts:

- [Networking Environments](#)
- [Net8 and the Transparent Network Substrate \(TNS\)](#)
- [Stack Communications](#)
- [Stack Communications in a Typical Oracle Networking Environment](#)
- [Stack Communications in an IIOP Environment](#)
- [Server-to-Server Interaction](#)
- [Stack Communications in a JDBC Environment](#)

Networking Environments

Oracle networking environments are based on two concepts:

- [Distributed Processing](#)
- [Distributed Databases](#)

Distributed Processing Oracle databases and client applications operate in what is known as a distributed processing environment. Distributed or cooperative processing involves interaction between two or more computers to complete a single data transaction. Applications such as an Oracle tool act as clients requesting data to accomplish a specific operation. Database servers store and provide the data.

In a typical network configuration, clients and servers may exist as separate logical entities on separate physical machines. This configuration allows for a division of labor where resources are allocated efficiently between a client workstation and the server machine. Clients normally reside on desktop computers with just enough memory to execute user friendly applications, while a server has more memory, disk storage, and processing power to execute and administer the database.

Distributed Databases This type of client-server architecture also enables you to distribute databases across a network. A distributed database is a network of databases stored on multiple computers that appears to the user as a single logical database. Distributed database servers are connected by a database link, or path from one database to another. One server uses a database link to query and modify information on a second server as needed, thereby acting as a client to the second server.

Net8 and the Transparent Network Substrate (TNS)

Net8 uses the Transparent Network Substrate (TNS) and industry-standard networking protocols to accomplish its basic functionality. TNS is a foundation technology that is built into Net8 providing a single, common interface to all industry-standard protocols.

With TNS, peer-to-peer application connectivity is possible. In a peer-to-peer architecture, two or more computers (called *nodes* when they are employed in a networking environment) can communicate with each other directly, without the need for any intermediary devices. In a peer-to-peer system, a node can be both a client and a server.

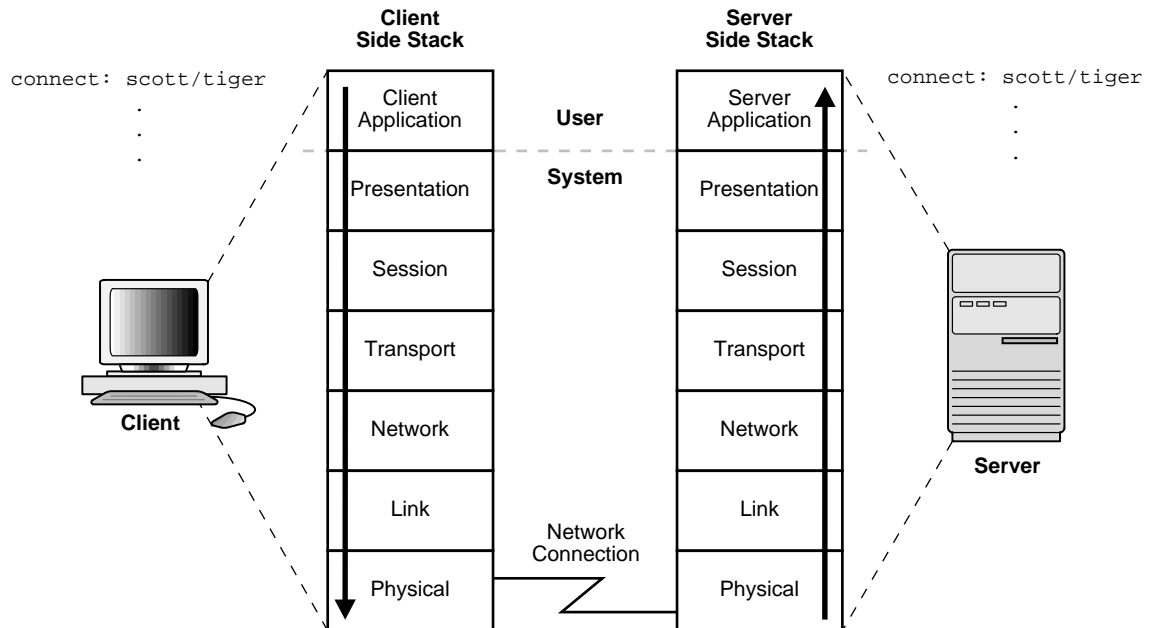
A review of how Oracle clients and servers operate and communicate with each other will help you to understand what TNS is and how it works with Net8 to establish network sessions.

Stack Communications

The concept of distributed processing relies on the ability of computers separated by both design and physical location to communicate and interact with each other. This is accomplished through a process known as stack communications.

Stack communications can be explained by referencing the Open System Interconnection (OSI) model. In the OSI model, communication between separate computers occurs in a stack-like fashion with information passing from one node to the other through several layers of code. [Figure 2-1](#) depicts a typical OSI Protocol Communications Stack.

Figure 2-1 OSI Communications Stack



Information descends through layers on the client side where it is packaged for transport across a network medium in a manner that it can be translated and understood by corresponding layers on the server side.

A typical OSI protocol communications stack will contain seven such layers:

client application The OSI layer closest to the user, and as such is dependent on the functionality requested by the user. For example, in a database environment, a Forms application may attempt to initiate communication in order to access data from a server.

presentation Ensures that information sent by the application layer of one system is readable by the application layer of another system. This includes keeping track of syntax and semantics of the data transferred between the client and server. If necessary, the presentation layer translates between multiple data representation formats by using a common data format.

session Establishes, manages, and terminates network sessions between the client and server. This is a virtual pipe that carries data requests and responses. The session layer manages whether the data traffic can go in both directions at the same time (referred to as asynchronous), or in only one direction at a time (referred to as synchronous).

transport Implements the data transport ensuring that the data is transported reliably.

network Ensures that the data transport is routed through optimal paths through a series of interconnected subnetworks.

link Provides reliable transit of data across a physical link.

physical Defines the electrical, mechanical, and procedural specifications for activating, maintaining and deactivating the physical link between client and server.

Stack Communications in a Typical Oracle Networking Environment

Stack communications allow Oracle clients and servers to share, modify, and manipulate data between themselves. The layers in a typical Oracle communications stack are similar to those of a standard OSI communications stack. This section covers the following topics:

- [Net8 Client-Side Interaction](#)
- [Server-Side Interaction](#)

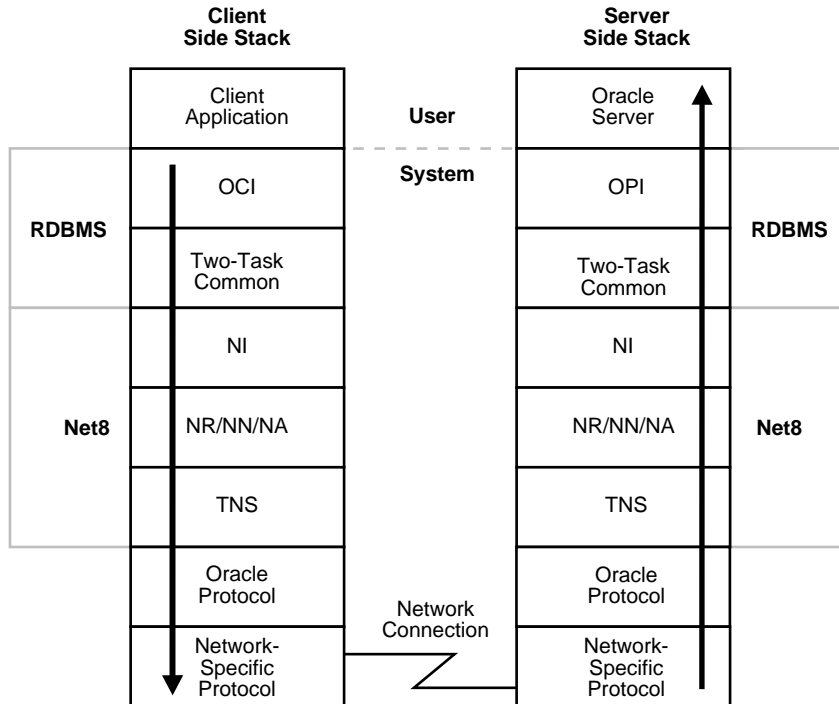
Net8 Client-Side Interaction

In an Oracle client-server transaction, information passes through the following layers:

- [Client Application](#)
- [Oracle Call Interface \(OCI\)](#)
- [Two-Task Common](#)
- [Net8](#)
- [Oracle Protocols](#)
- [Oracle Protocols](#)
- [Network-Specific Protocols](#)

Figure 2-2 depicts a typical communications stack in an Oracle networking environment.

Figure 2-2 Typical Communications Stack in an Oracle environment



Client Application Oracle client applications provide all user-oriented activities, such as character or graphical user display, screen control, data presentation, application flow, and other application specifics. The application identifies database operations to send to the server and passes them through to the Oracle Call Interface (OCI).

Oracle Call Interface (OCI) The OCI code contains all the information required to initiate a SQL dialogue between the client and the server. It defines calls to the server to:

- parse SQL statements for syntax validation
- open a cursor for the SQL statement
- bind client application variables into the server shared memory
- describe the contents of the fields being returned based on the values in the server's data dictionary
- execute SQL statements within the cursor memory space
- fetch one or more rows of data into the client application
- close the cursor

The client application uses a combination of these calls to request activity within the server. OCI calls can be combined into a single message to the server, or they may be processed one at a time through multiple messages to the server, depending on the nature of the client application. Oracle products attempt to minimize the number of messages sent to the server by combining many OCI calls into a single message to the server. When a call is performed, control is passed to Net8 to establish the connection and transmit the request to the server.

For more information about OCI, see the *Oracle8i Call Interface Programmer's Guide*.

Two-Task Common Two-Task Common is Oracle's implementation of the [presentation](#) layer. Two-Task Common provides character set and data type conversion between different character sets or formats on the client and server. This layer is optimized to perform conversion only when required on a per connection basis.

At the time of initial connection, Two Task Common is responsible for evaluating differences in internal data and character set representations and determining whether conversions are required for the two computers to communicate.

Net8 Net8 provides all [session](#) layer functionality in an Oracle communications stack. It is responsible for establishing and maintaining the connection between a client application and server, as well as exchanging messages between them. Net8 itself has three component layers that facilitate session layer functionality:

Component	Description
Network Interface (NI)	This layer provides a generic interface for Oracle clients, servers, or external processes to access Net8 functions. The NI handles the "break" and "reset" requests for a connection
Network Routing (NR)/ Network Naming (NN)/ Network Authentication (NA)	NR provides routing of the network session to the destination. This may include any intermediary destinations or "hops" on the route to the server destination. NN resolves net service names to a Net8 destination address. NA negotiates any authentication requirement with the destination.
Transparent Network Substrate (TNS)	TNS is an underlying layer of Net8 providing a common interface to industry standard protocols. TNS receives requests from Net8, and settles all generic machine-level connectivity issues, such as: the location of the server or destination (open, close functions); whether one or more protocols will be involved in the connection (open, close functions); and how to handle interrupts between client and server based on the capabilities of each (send, receive functions). The generic set of TNS functions (open, close, send, receive) passes control to an Oracle protocol to make a protocol-specific call. Additionally, TNS supports encryption and sequenced cryptographic message digests to protect data in transit. TNS is the basic component of the TNS Network session (NS) layer.

Oracle Protocols Oracle protocols are Oracle's implementation of the [transport](#) layer. Oracle protocols are responsible for mapping TNS functionality to industry-standard protocols used in the client-server connection. Each protocol is responsible for mapping the equivalent functions between TNS and a specific protocol. Oracle protocols include:

- LU6.2
- Named Pipes
- SPX
- TCP/IP
- TCP/IP with SSL

Network-Specific Protocols All Oracle software in the client-server connection process require an existing network protocol stack to make the machine-level connection between the two machines for the **transport** layer. The network protocol is responsible only for getting the data from the client machine to the server machine, at which point the data is passed to the server-side Oracle protocol.

Server-Side Interaction

Information passed from a client application across a network protocol is received by a similar communications stack on the server side. The process stack on the server side is the reverse of what occurred on the client side with information ascending through communication layers. The one operation unique to the server side is the act of receiving the initial connection through the listener.

The following components above the network session layer are different from those on the client side:

- Oracle Program Interface (OPI)
- Oracle Server

Oracle Program Interface (OPI) The OPI performs a complementary function to that of the OCI. It is responsible for responding to each of the possible messages sent by the OCI. For example, an OCI request to fetch 25 rows would have an OPI response to return the 25 rows once they have been fetched.

Oracle Server The Oracle Server side of the connection is responsible for receiving dialog requests from the client OCI code and resolving SQL statements on behalf of the client application. Once received, a request is processed and the resulting data is passed to the OPI for responses to be formatted and returned to the client application.

Server-to-Server Interaction

When two servers communicate to complete a distributed transaction, the process, layers, and dialogues are the same as in the client-server scenario, except that there is no client application. The server has its own version of OCI, called the Network Program Interface (NPI). The NPI interface performs all of the functions that the OCI does for clients, allowing a coordinating server to construct SQL requests for additional servers.

Stack Communications in an IIOB Environment

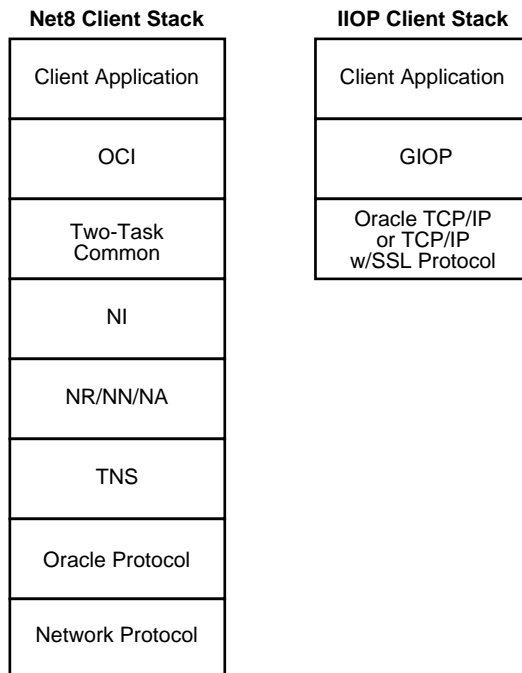
This release of Net8 supports a new **presentation** layer called General Inter-Orb Protocol (GIOP) that is used for those clients connecting to the Java option. Internet Inter-Orb Protocol (IIOB) is an implementation of GIOP over TCP/IP or TCP/IP with SSL. Oracle provides the GIOP service implementation.

IIOB Client-Side Interaction

IIOB clients have different communication stack than a typical Net8 client stack, as shown in **Figure 2-3**. IIOB clients use:

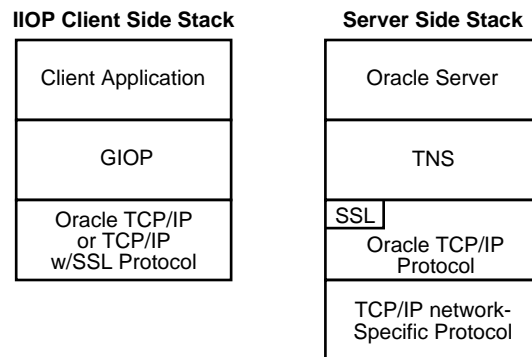
- GIOP as the presentation layer rather than **Two-Task Common**
- no **session** layering rather than **Transparent Network Substrate (TNS)**

Figure 2-3 Typical Client vs. IIOB Client Communication Stacks



The server side does not require many of the Net8 communication layers needed in a typical Net8 connection. Instead, the server side only requires a network protocol of TCP/IP and an Oracle protocol of TCP/IP or TCP/IP with SSL. The only component of Net8 required is [Transparent Network Substrate \(TNS\)](#). [Figure 2-4](#) depicts a communication stack between an IIOP client and the Java option in the Oracle database.

Figure 2-4 Client/Server Communications Stack in an IIOP Environment



See "[Configuring Java Option Connections](#)" on page 7-50.

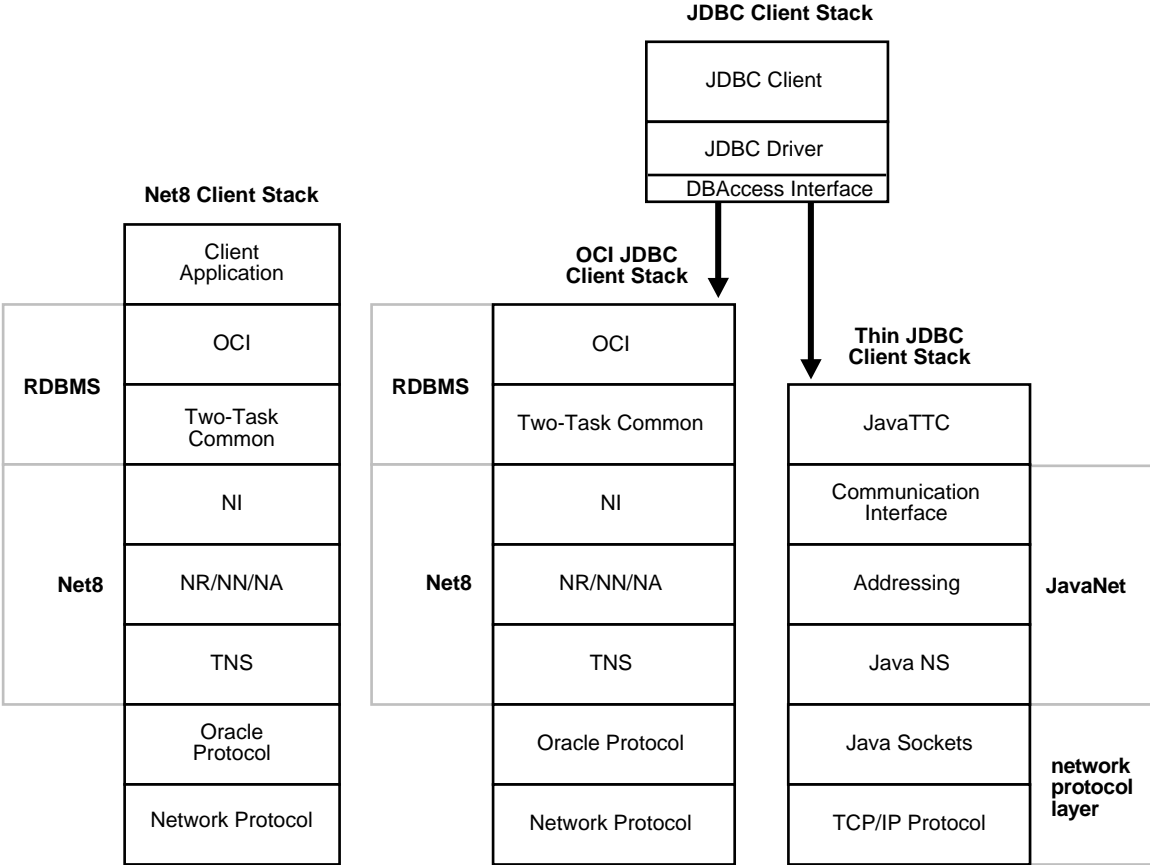
Stack Communications in a JDBC Environment

Oracle's Java DataBase Connectivity (JDBC) Drivers provide Java programmers access to an Oracle database. Oracle offers two JDBC drivers—one targeted specifically to Java developers writing client-server Java applications and Java-based middle tiers (Java web servers and Java application servers), the other targeted those developing Java applets.

- JDBC/OCI is Oracle's Level 2 JDBC driver is targeted to client-server Java application programmers and Java-based middle-tier developers. The JDBC/OCI driver converts JDBC invocations to calls to the Oracle Call Interface (OCI) which are then sent over Net8 to the Oracle database server.
- Thin JDBC driver is Oracle's Type 4 driver designed for Java applet and Java application developers. The JDBC driver establishes a direct connection to the Oracle database server over Java sockets. Access to the database is assisted with a lightweight implementation of [Net8](#) and [Two-Task Common](#).

The layers in a JDBC client communications stack are similar to those of a standard Oracle communications stack, as shown in [Figure 2-5](#) on page 2-17:

Figure 2-5 Typical Client vs. JDBC Client Communication Stacks



The OCI JDBC client stack is similar to a typical Net8 Client stack. The Thin JDBC client stack has similar components implemented differently. In an Java client applet-server transaction, information passes through the following layers:

- [DBAccess Interface](#)
- [JavaTTC](#)
- [JavaNet](#)
- [Network Protocol Layer](#)

DBAccess Interface

The Java DBAccess Interface layer allows the client to initiate a SQL session much like [Oracle Call Interface \(OCI\)](#) does in a standard client communication stack.

JavaTTC

Provides a subset version of the [Two-Task Common](#) implementation necessary for exchanges of information between the Java client and the database. JavaTTC is responsible for:

- negotiating protocol version and data type
- determining whether conversions between the Java client character set and server character set are necessary
- SQL statement execution

JavaNet

Provides a communication infrastructure so JDBC client can connect to an Oracle database that is an equivalent implementation of NI, NR/NN/NA, and TNS in Net8.

- Communication Interface - Provides an interface between [JavaTTC](#) and other networking layers much like NI does in a standard client communication stack.
- Addressing - Processes Net8 destination addresses much like NN does in a standard client communication stack.
- Java NS - Much like [Transparent Network Substrate \(TNS\)](#) in a standard client communication stack, Java NS provides a common interface to TCP/IP, settles all generic machine-level connectivity issues, such as: the location of the server or destination (open, close functions); whether one or more protocols will be involved in the connection (open, close functions); and how to handle interrupts between client and server based on the capabilities of each (send, receive functions).

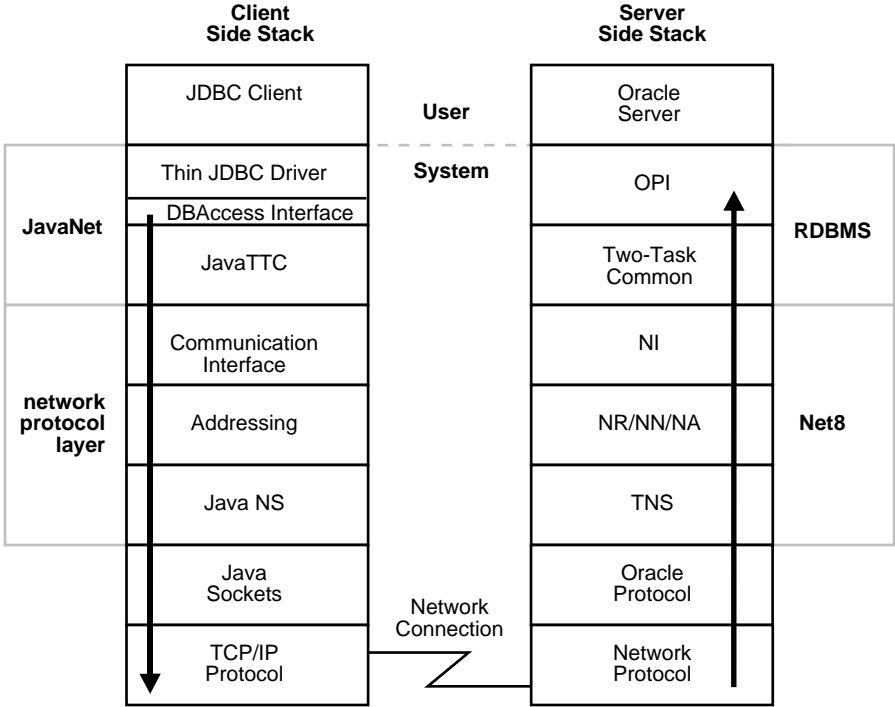
Network Protocol Layer

Java Sockets - Provides a connection to between Java NS and TCP/IP.

TCP/IP Protocol - Java NS will only run on top of TCP/IP, because Java sockets use TCP/IP.

Figure 2-6 depicts a communication stack between a Java client applet and an Oracle database.

Figure 2-6 Client/Server Communications Stack in a JDBC Environment



Network Sessions to Multi-Threaded Servers

Note: To take advantage of multi-threaded server (MTS), you must configure it, as described in [Chapter 9, "Configuring Multi-Threaded Server"](#).

Multi-threaded server (MTS), also called *shared server*, allows many user processes to share very few server processes.

With MTS configuration, many user processes connect to a dispatcher. A *dispatcher* enables many clients to connect to the same server without the need for a dedicated server for each client. A dispatcher handles and directs multiple incoming network session requests to the shared server. The advantage of MTS is that system overhead is reduced and less resources are used, allowing the number of users supported to be increased.

Without MTS, each user process requires its own dedicated server process. A new server process is created for each client requesting a connection. A dedicated server remains associated to the user process for the remainder of the connection.

When an Oracle server has been configured as MTS, incoming network sessions are always routed to the dispatcher unless either the network session specifically requests a dedicated server or no dispatchers are available. The sequence of events that occurs with the dispatcher server is as follows:

1. The listener is started and listens on either a default address or the addresses specified in its configuration file.
2. A database instance starts. Dispatchers start according to the configuration parameters in the initialization parameter file. Each dispatcher then performs a listen on the address assigned to it.
3. Each dispatcher's address is registered with the listener. When the listener is not listening on its default address, the listener's network name may be specified in the database initialization file (INITSID.ORA). The name may resolve to more than one such address if multiple listeners are used.

Once the dispatcher addresses are registered, the listener can redirect incoming connect requests to them.

If step 2 is performed before step 1, the server will not be able to contact the listener in step 3. If this occurs, there may be a delay as the server attempts to connect to the listener. If a connect request comes in a timeframe where no dispatchers are

registered, these requests may either be handled through prespawnd dedicated or newly spawned dedicated servers or may be rejected.

The listener and the Oracle dispatcher server are now ready to receive incoming network sessions.

Note: You can check which dispatchers have registered with the listener by issuing a SERVICES command in the Listener Control Utility. For more information, see "SERVICES" on page A-10.

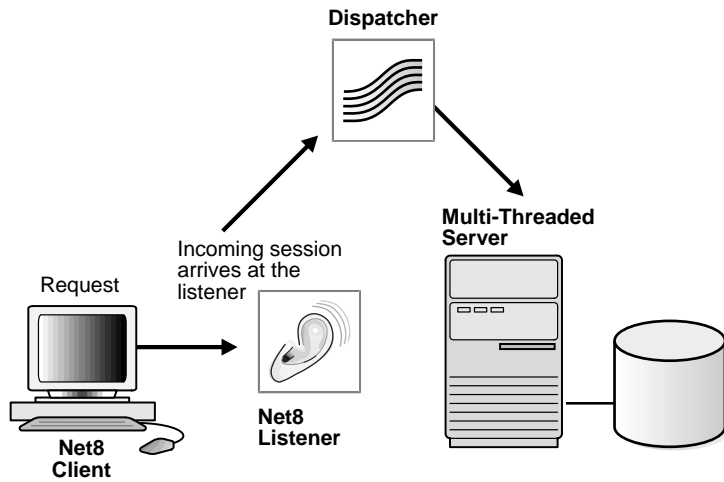
Once the listener and the dispatcher server have been started, the network session activity continues as follows:

1. The client connects to the listener with the network address.
2. The listener receives the connect request, and determines if the client's request may be serviced. If not, the listener refuses the network session, and then resumes at Step 6.
3. If the client's request is validated, the listener issues a redirect message to the client containing the network address of the least-used dispatcher for the shared server. In Oracle Parallel Server, the message would be redirected to the least used dispatcher on the least loaded node.
4. The client dissolves the network session to the listener and establishes a network session to the shared server using the network address provided in the redirect message.
5. The dispatcher updates the listener with the new load value because of the presence of the new network session. This allows the listener to balance the incoming network session requests between dispatchers running on the same protocol.
6. The listener resumes listening for incoming network sessions.

When a client disconnects, the shared server associated with the client stays active and processes other incoming requests. Different requests from the same client may be processed by different shared servers.

Figure 2-7 depicts the role of the listener in a redirected connection to a dispatcher.

Figure 2-7 *Redirected Connection to a Dispatcher*



For MTS configuration information, see [Chapter 9, "Configuring Multi-Threaded Server"](#).

Network Sessions to Dedicated Servers

Alternatively to MTS, Net8 may redirect the request to an existing *dedicated server*. *This is the default implementation*. Net8 sends the address of an existing server process back to the client. The client then resends its connect request to the server address provided.

Net8 provides the option of automatically creating dedicated servers *before* the request is received. These processes last for the life of the listener, and can be reused by subsequent connection requests. The use of prespawnd dedicated server processes requires specification in a listener configuration file.

Note: Prespawnd dedicated servers require SQL*Net version 2.1 or later, and Oracle Server release 7.1 or later.

The sequence of events that occurs when using prespawnd dedicated server processes to service client connection requests is as follows:

1. The listener is started and listens on an address specified in a listener configuration file.
2. The listener then spawns a series of dedicated server processes until it reaches the specified pool size defined in its configuration file.
3. Each spawned server process performs a partial address listen and provides the listener with the partial address that it is listening on. The listener initially marks all prespawnd servers as idle.

Note: In a partial address listen, the server process listens, but informs the underlying protocol stack that it has no preference as to the specific address it will listen on. As a result, many protocol stacks will choose a free listening address and automatically assign this to the requesting server process.

4. The client sends a connect request to the listener.
5. The listener receives the network session request, and determines if the client's request may be serviced. If not, the listener refuses the network session and then resumes at Step 9.

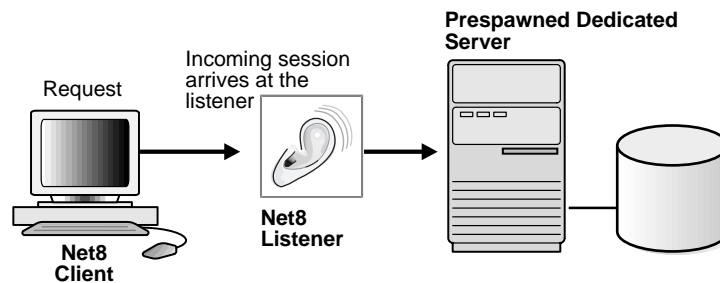
6. The listener issues a redirect message to the client containing one of the network addresses of the prespawned servers. The listener logs that server as active.
7. The client dissolves the network session to the listener and establishes a network session to the prespawned server using the address provided in the redirect message.
8. The listener spawns another server process to replace the active prespawned server (provided a value called `PRESPAWN_MAX` in the listener configuration file is greater than the number of prespawned server processes active and idle).
9. The listener continues listening for incoming network sessions.

The above sequence of events continues until the maximum prespawn limit is reached, at which point the listener stops spawning new dedicated server processes.

When clients disconnect, the prespawned dedicated server process associated with the client returns to the idle pool. It then waits a specified length of time to be assigned to another client. If no client is handed to the prespawned server before the timeout expires, the prespawned server shuts down.

[Figure 2–8](#) depicts the role of the listener in a redirected connection to a prespawned dedicated server process.

Figure 2–8 *Redirected Connection To a Prespawned Dedicated Server Process*



Bequeathed Network Sessions to Dedicated Servers

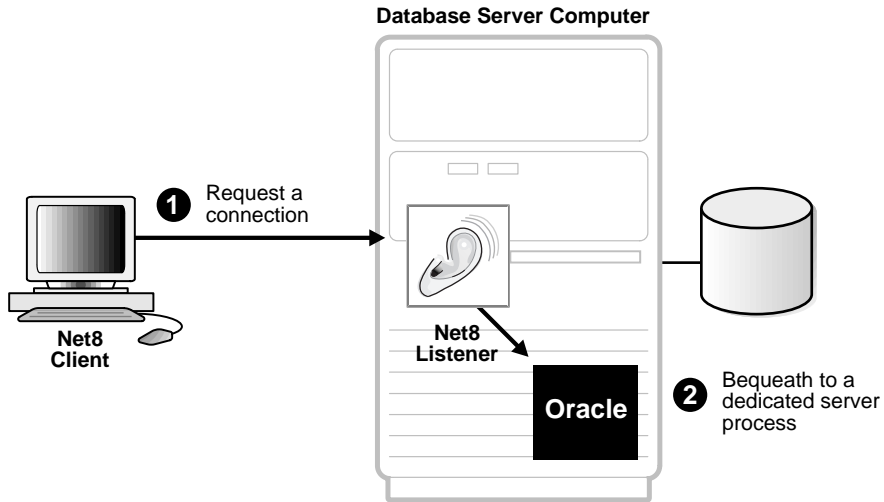
If the listener and server exist on the *same* node, the listener may create or spawn dedicated servers as connect requests are received. Dedicated servers are committed to one network session only and exist for the duration of that network session. The sequence of events that occur when the listener creates a dedicated server process and passes or "bequeaths" control of a network session to it is as follows:

1. The listener is started and listens on an address specified in a listener configuration file (LISTENER.ORA).
2. A client connects to the listener with the network address.
3. The listener receives the network session request, and determines if the client's request may be serviced. If not, the listener refuses the network session and then resumes at Step 5.
4. The listener spawns a new dedicated server process to serve the incoming network session, and bequeaths the network session to that server process. Once the network session is established, data flows directly between the client and dedicated server process.
5. The listener continues listening for incoming network sessions.

When a client disconnects, the dedicated server process associated with the client closes.

Figure 2–9 depicts the role of the listener in a bequeathed connection to a dedicated server process.

Figure 2–9 *Bequeathed Connection To a Dedicated Server Process*



Net8 and the Listener

The listener receives connection requests on behalf of a client application.

This section covers the following topics:

- [Database Instance Registration to Listeners](#)
- [Client Connections](#)
- [Connect-Time Failover](#)
- [Load Balancing](#)

Database Instance Registration to Listeners

Database instances register themselves with the listener when started. Database instance registration is comprised of two elements:

- service registration provides the listener with instance information, such as database service names and instance names.
- MTS dispatcher registration provides dispatcher information to the listener

When an instance is started, initialization parameters are read from the `INITSID.ORA`. One of these initialization parameters is the service name. By default, an instance background process registers instance information to a listener on the local machine.

If a listener is started after the instance, there may be a delay before the instance and dispatchers are registered. The instance will attempt to connect to the listener periodically. Similarly, if a listener gets an incoming request before an instance is registered, the listener may reject the request.

For MTS configuration information, see [Chapter 9, "Configuring Multi-Threaded Server"](#).

Benefits

Database instance registration offers the following benefits:

- The `LISTENER.ORA` file does not require the `SID_LIST_listener_name` parameter that specifies information on the databases served by the listener. This parameter is still required if the management tool you are using still requires it.
- Connect-time failover is enabled, as described in "[Connect-Time Failover](#)" on page 2-29.
- Connection load balancing is enabled, as described in "[Load Balancing](#)" on page 2-30.

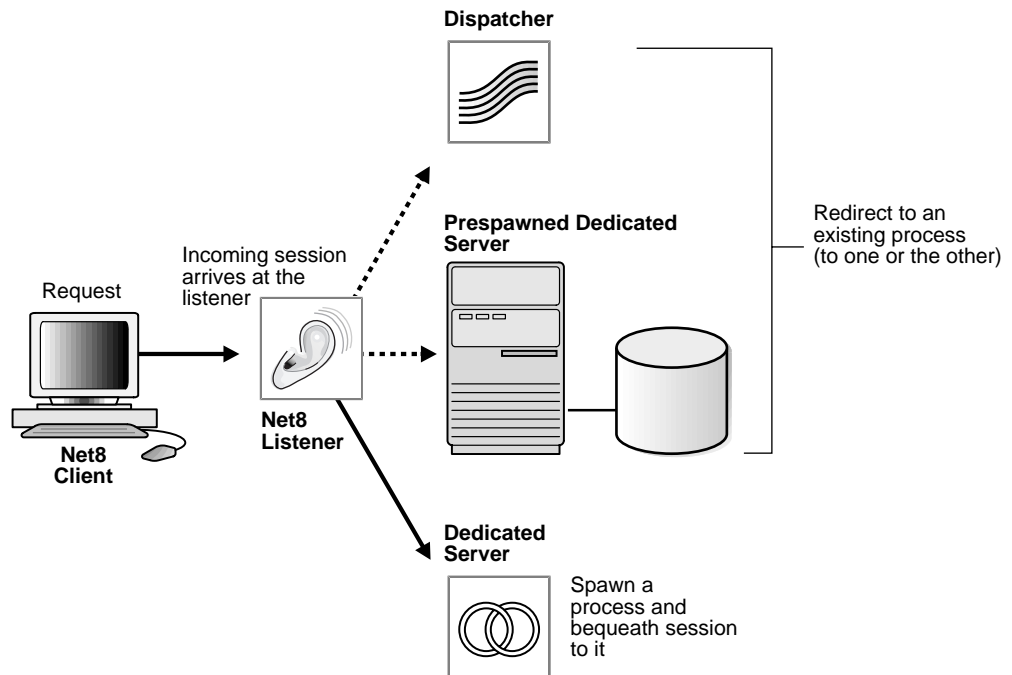
Client Connections

A client is configured with a net service name that contains the location of a listener and service name of the service to which to connect.

When a client requests a connection from a listener, the service name is sent to the listener. The listener receives the network session request, determines if the information passed to it matches the information it has registered, and determines if the client request may be serviced.

Figure 2–10 depicts the role of the listener in a typical Net8 connection to a server.

Figure 2–10 Listener In a Typical Net8 Connection



Connect-Time Failover

Database instance registration enables the listener to know if an instance is up prior to attempting a connection. This feature facilitates automatic failover of a client connect request to a different listener.

To control how the client executes these connection attempts, you will need to configure multiple listening addresses for each net service name and use the failover (FAILOVER = ON) parameter.

Connect-time failover continues until the client successfully connects to a listener.

CAUTION: Implementing connect-time failover does not allow use of static service configuration parameters in the LISTENER.ORA file. However, static configuration is required for Oracle8i release 8.0 or Oracle7 databases and Oracle Enterprise Manager. For further information about statically configuring the listener, see ["Configuring a Listener with Service Information"](#) on page 6-71.

See the *Oracle8i Parallel Server Setup and Configuration Guide* for additional configuration instructions on setting up connect-time failover for Oracle Parallel Server in an Oracle Enterprise Manager environment.

Connect-time failover is not the same as "Transparent Application Failover" which is a runtime-failover for high-availability environments, such as Oracle Parallel Server, that refers to the failure of and re-establishing of application-to-service connections. See *Oracle8i Tuning* for further information about transparent application failover.

For configuration information, see ["Configuring Multiple Address Options"](#) on page 7-11.

Load Balancing

Load balancing is a feature by which client connections are distributed evenly among multiple listeners, dispatchers, instances, and nodes so that no single component is overloaded.

Load balancing happens at the following levels:

- [Client Load Balancing](#)
- [Connection Load Balancing](#)

Client Load Balancing

If more than one listener services a single database, a client can randomly choose between the listeners for its connect requests. This randomization allows all listeners to share the burden of servicing incoming connect requests.

To enable your clients to choose from listeners at random, you will need to configure multiple listening addresses for each net service name and use the load balance (`LOAD_BALANCE=ON`) parameter.

For additional configuration information, see "[Configuring Multiple Address Options](#)" on page 7-11.

Connection Load Balancing

Database instance registration enables connection load balancing because of the registration that happens with remote listeners. Connection load balancing evenly distributes the number of active connections among various instances and dispatchers for the same service. The load of an instance and dispatcher is determined by the number of connections.

Connection load balancing is only enabled for an MTS environment. It is not configurable by clients.

A listener sends an incoming client request for a specific service to the least loaded dispatcher and instance. For example, if the service has multiple instances on multiple nodes, it chooses a dispatcher based on the least loaded instance, where the instance load is based on the node load.

Oracle Names Architecture

Oracle Names is a distributed naming service developed for Oracle environments to help simplify the setup and administration of global, client/server computing networks.

This section covers the following topics:

- [Overview](#)
- [Administrative Regions](#)
- [Domains](#)
- [Oracle Names As Data Repository](#)
- [Data Stored in an Oracle Names Server](#)
- [Organizing and Naming Network Components](#)
- [Using Multiple Regions to Decentralize Administrative Responsibilities](#)
- [Differences Between Versions of Oracle Names](#)
- [Understanding Discovery](#)

Overview

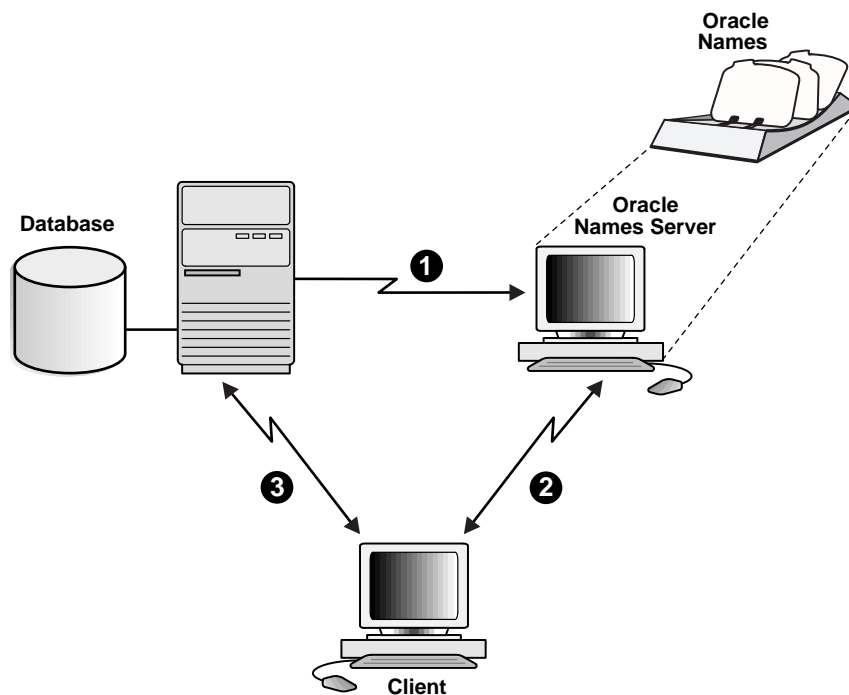
Oracle Names establishes and maintains an integrated system of Oracle Names servers which work together like a directory service storing addresses for all the services on a network and making them available to clients wishing to make a connection.

Much like a caller who uses directory assistance to locate a telephone number, clients configured to use Oracle Names will refer their connection requests to a *Oracle Names server*. The Oracle Names server will attempt to resolve the service name provided by the client to a network address. If the Oracle Names server finds the network address, it will then return that information to the client. The client can then use that address to connect to the service.

Figure 2–11 depicts how Oracle Names works to help establish a connection between a client and server:

1. A database registering its service with an Oracle Names server.
2. Client seeking to locate a service on the network by contacting an Oracle Names server to retrieve the network address.
3. Clients then transparently connecting to the service:

Figure 2–11 Oracle Names



Oracle Names provides an alternative to file-based or local name resolution methods, where net service names and addresses must be configured and maintained with each individual client. By maintaining this information in a central administrative location, Oracle Names reduces the work effort associated with adding or relocating services.

For configuration information, see "[Configuring the Network with the Oracle Names Method](#)" on page 6-21.

Administrative Regions

Oracle Names provides support for one or more administrative regions.

Most networks have one central point of administration, that is, one *administrative region*. An administrative region consists of a collection of Oracle Names servers that administer services in a network. All connect information are stored in a single data repository, which has the authority to interpret a service name. All Oracle Names servers within an administrative region query information from this data repository. If the administrative region uses a database for storage, there is one database per administrative region. There can be any number of Oracle Names servers.

Most enterprise environments with multiple data centers and many Oracle instances will probably choose to take advantage of multiple administrative regions. This allows each data center to independently define and manage the services in its own environment. At the same time, all service addresses are continuously available to all of the clients in the whole environment. Oracle Names servers transparently forward name resolution requests from clients in foreign administrative regions to the proper Oracle Names server.

Domains

An administrative region contains one or more domains used to divide administrative responsibilities.

A *domain* is a logical group of machines and network services. A domain is a way to identify a service within a network. Similar to a directory structure in a file system, it assigns a unique logical name to a network service. Within each domain all names must be unique, but across domains simple unqualified names can be repeated.

Network domains are similar to file directories used by many operating systems in that they are hierarchical. Unlike file systems however, network domains may or may not correspond to any physical arrangement of databases or other objects in a network. They are simply names spaces developed to prevent name space conflicts.

Note: Although they appear similar, the domains of an Oracle network are completely independent of Domain Name Service (DNS) name spaces. For convenience, you may choose to mirror the DNS directory structure in your Oracle network.

Oracle Names As Data Repository

Data in Oracle Names servers is updated through continuous replication between all the Oracle Names servers in the region, or by writing to and reading from a common Oracle database.

For smaller workgroup environments where all of the services are registered dynamically, administrators may configure Oracle Names servers to replicate data continuously among themselves. When a listener registers a new service, information about that service will immediately be passed along to other Oracle Names servers in the administrative region.

Alternatively, administrators in large environments will normally want to store their registration data in an Oracle database, called the *region database*. A region database consists of tables that store Oracle Names information. If the Oracle Names servers are configured to use an Oracle database as a repository, all service registrations will be written to the database. Each Oracle Names server in a given administrative region will periodically poll the region database for updated registrations. In this way, new registrations are communicated in a timely manner to all of the Oracle Names servers in a given administrative region. At the same time, it relieves Oracle Names servers of the necessity to communicate directly with each other, as well as provides better reliability.

Data Stored in an Oracle Names Server

Below is a description of the types of data stored in an Oracle Names server.

Data	Description
database global database names and addresses	The Oracle Names server retrieves information about the database, including the global database name (database name and domain) and address, from the listener. The address is configured in the LISTENER.ORA file, and the global database name is registered during database startup or statically configured in the LISTENER.ORA file. You do not to register this information.
other Oracle Names server names and addresses	An Oracle Names server stores the names and addresses of all other Oracle Names servers in the same administrative region. If there is more than one administrative region in a network, the Oracle Names server will store the name and address of at least one Oracle Names server in the root administrative region and each of the immediate sub-regions. You do not need to register this information.
net service names	If you register net service name with the NAMESCTL control utility or the Net8 Assistant, an Oracle Names server stores them. An Oracle Names server also stores gateways to non-Oracle databases and Oracle RDB databases.

Data	Description
global database link	<p><i>Database links</i> allow a database to communicate with another database another. The name of a database link is the same as the global database name of the database to which the link points. Typically, only one database link should exist per database.</p> <p>The following types of database links may be created:</p> <ul style="list-style-type: none"> ■ You can create a <i>private database link</i> in a specific schema of a database. Only the owner of a private database link can use it. ■ You can create a <i>public database link</i> for a database. All users and in the database can use it. ■ You can create a <i>global database link</i> in Oracle Names server so anyone in the network can use it. <p>Because the Oracle Names retrieves the global database name from the listener, a global database link that is the global database name is automatically registered with the Oracle Names server. Therefore, you do not need to register this information.</p> <p>User name and password credentials for the global database link may be registered with the Oracle Names server using the Net8 Assistant. These global database links may be supplemented with link qualifiers defined through the Net8 Assistant.</p> <p>Global database links may be superseded with private and public database links created by individual users. For more information about private and public database links, see <i>Oracle8i Distributed Database Systems</i>.</p>
aliases	<p>An Oracle Names server stores aliases or alternative service names for any defined net service name, database service or global database link. Aliases may be registered with the Oracle Names server using either the NAMESCTL control utility or Net8 Assistant.</p>
Oracle Connection Managers	<p>An Oracle Names server stores the names and listening addresses of all Oracle Connection Managers on the network. You do not to register this information.</p>

Organizing and Naming Network Components

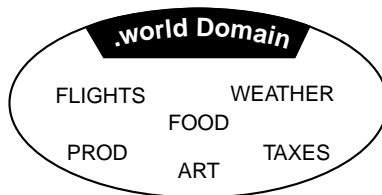
When you use Oracle Names, objects such as databases in a networked environment will need to be named in a way as to ensure that they are unique within the network. There are two basic models for naming objects in a network:

- [Single Domain Model](#)
- [Hierarchical Naming Model](#)

Single Domain Model

The use of the single domain naming model is useful if your network is small, and there is no duplication of names. [Figure 2–12](#) depicts a typical flat naming structure using a single domain name, .WORLD.

Figure 2–12 Single Domain Naming Model



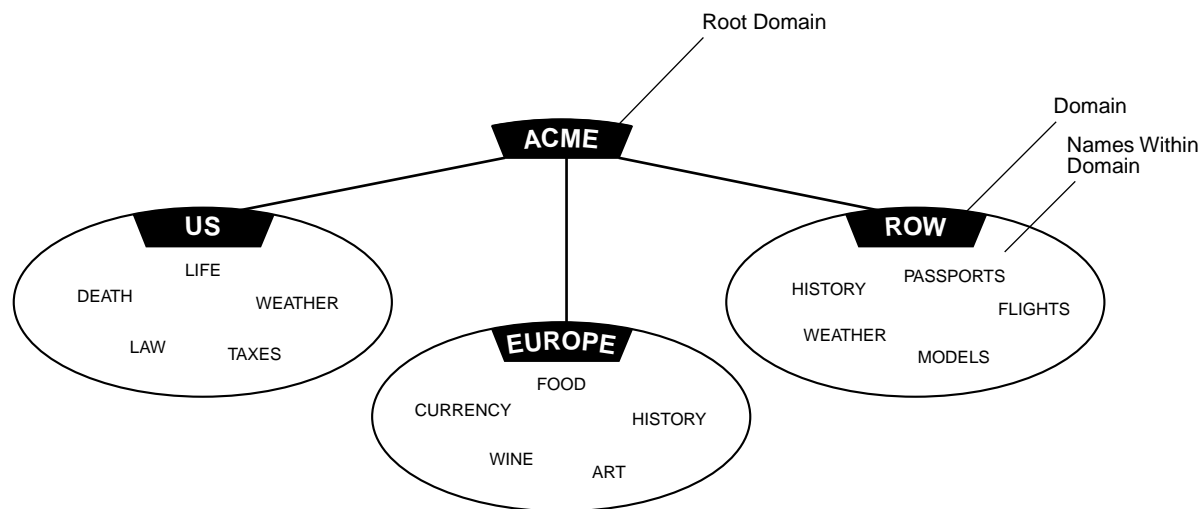
In this environment, database service names will automatically be appended with a .WORLD extension (for example, PROD.WORLD, FLIGHTS.WORLD, and so forth).

Hierarchical Naming Model

Hierarchical naming models divide names into a hierarchical structure to allow for future growth or greater naming autonomy. This type of naming model will allow more than one database with the same simple name in different domains.

Figure 2–13 depicts a hierarchical structure of domains including the (ROOT) domain, ACME domain, US.ACME, EUROPE.ACME, and ROW.ACME (Rest of World) domains.

Figure 2–13 Hierarchical Naming Model

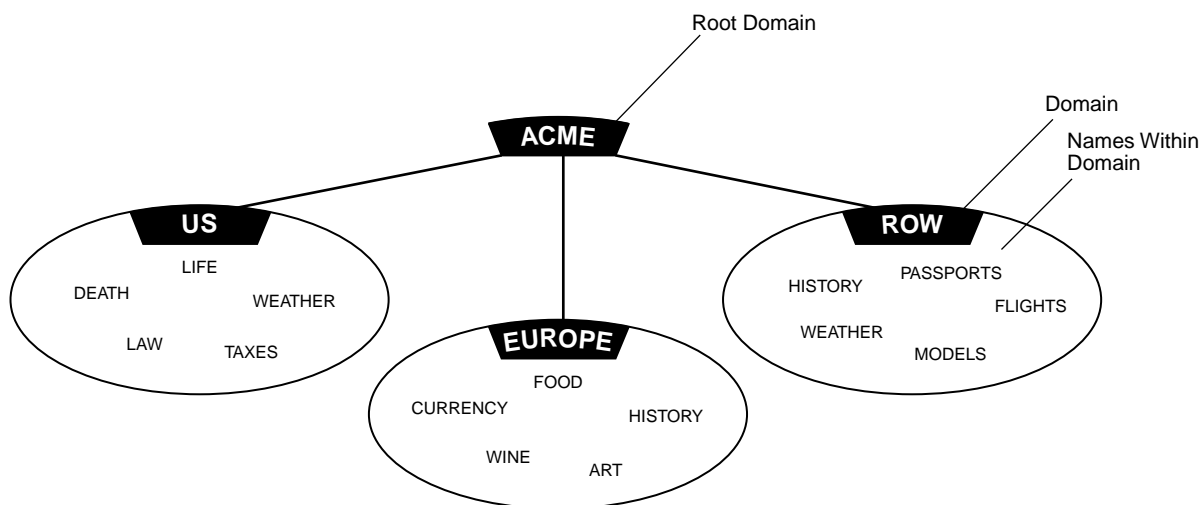


Notice in Figure 2–13 both WEATHER and HISTORY are repeated, but the names remain unique (that is, HISTORY.ROW.ACME and HISTORY.EUROPE.ACME).

Default Domains The default domain is the domain within which most of the client's name requests are conducted. This is usually the domain in which the client resides, though it could also be another domain from which the client most often requests services. A client can request a network service within its default domain using the service's simple, unqualified name, that is, without specifying a domain name. If a user requests a name without a "." character in it, the default domain name is automatically appended to the database service or database link name requested.

Figure 2-14 depicts a client with a default domain of EUROPE.ACME.COM. When it makes a request for the service name "WINE", the default domain name EUROPE.ACME.COM is appended to the requested name so that the name becomes WINE.EUROPE.ACME.COM.

Figure 2-14 Default Domains



For more information about domain names, see *Oracle8i Concepts*.

Multiple Domains Multiple domains are related hierarchically to a root domain (the highest-level domain in the hierarchy) in a series of parent-child relationships. For example, under the root might be several domains, one of which is called COM. Under the COM domain might be several more domains, one of which is ACME. Under the ACME domain might be several domains, such as US, EUROPE, and so forth.

Note: In previous releases of SQL*Net and Oracle Names, a network with only one domain, would by default be called ".world". This is no longer a requirement with Net8 and Oracle Names version 8. You may, however, want to keep the same convention to be backward compatible, as well as to avoid having to rename all your databases.

Using Multiple Regions to Decentralize Administrative Responsibilities

An administrative region can contain multiple domains. A fully qualified domain name can exist in only one administrative region. However, a domain name can appear in two different regions. In order to have a flexible naming scheme, the administration responsibility of a region must be decentralized.

If you are using Oracle Names and your network is large or widely distributed geographically, you may choose to subdivide separate multiple regions. For example, if your network includes both the United States and Europe, you might want to have administrative decisions about the network made locally. To subdivide, you must delegate regions and domains from a parent to a child or subregion.

To delegate administrative regions, you must use a hierarchical naming model with each administrative region controlling one or more different domains.

Networks with multiple administrative regions are composed of one *root administrative region* and one or more *delegated administrative regions*.

Root Administrative Regions

The root administrative region contains the *root domain*. The root administrative region contains the following information:

- Oracle Names servers in the root region.
- Domains as they are administered in this region. This is always at least the root domain, and can include other domains.
- Delegated administrative region Oracle Names servers—The domains and Oracle Names server addresses in any alternate regions which act as direct child regions of the root.
- Data definitions for the root region—all of the database service names, database links and aliases associated with the root administrative region.

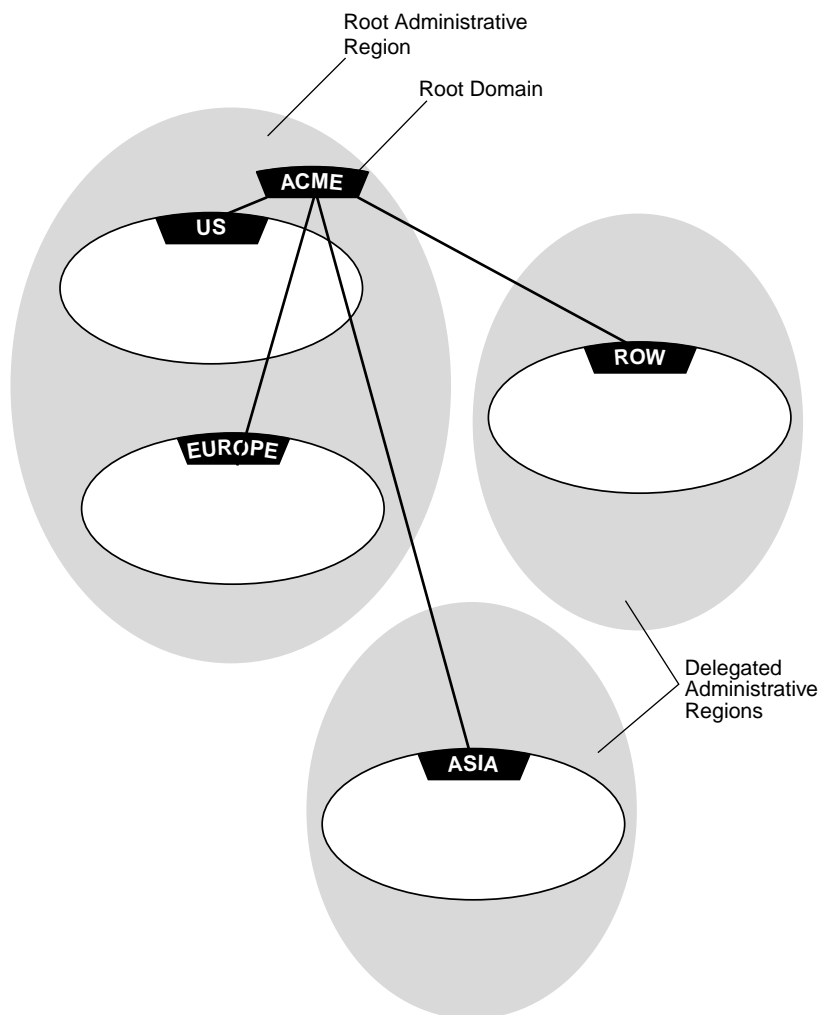
Delegated Administrative Regions

Administrative regions can be "delegated" from the top of the hierarchy down to other domains in the naming model. For example, a network with ten domains can have between one and ten administrative regions.

All administrative regions other than the root are hierarchically delegated directly or indirectly from it.

Figure 2-15 depicts a network with five domains and three administrative regions: the ROOT, and two delegated regions (DR1, DR2).

Figure 2-15 Delegated Administrative Regions



Delegated Administrative Regions Below Root

All administrative regions below the root are considered delegated administrative regions. Delegated administrative regions receive administrative responsibilities for a domain from other regions, such as the root administrative region. A delegated administrative region contains the following information:

- All Oracle Names servers and domains in the region
- Domains and Oracle Names server addresses in any of this administrative region's child regions
- Addresses of the Oracle Names servers in the root region. Having this data allows Oracle Names servers in delegated regions to contact any other region (through the root region)
- Data definitions—All of the database service names, database links, and aliases for all of the domains in this local (delegated) administrative region

Differences Between Versions of Oracle Names

There are significant differences between this version of Oracle Names and earlier versions:

- [Oracle Names version 1](#)
- [Oracle Names version 2](#)
- [Oracle Names version 8](#)

Oracle Names version 1

In Oracle Names version 1, administrators configured Oracle Names servers using Oracle Network Manager and stored all topology data in a database. All the Oracle Names servers in a region shared the same information because they accessed the same database.

The clients had a list of *preferred Oracle Names server* specified in the SQLNET.ORA file. This list was created by the user, listing the order of preferred Oracle Names to contact. The first Oracle Names server in the list would be contacted first by a client.

Preferred Oracle Names servers may still be configured, as described in "[Configuring Preferred Oracle Names Servers](#)" on page 7-20.

Oracle Names version 2

In Oracle Names version 2, the administrator could choose between continuing Oracle Names Server configuration as in version 1, or using the Dynamic Discovery Option. The Dynamic Discovery Option was recommended only for a network with a single region and single DNS domain. The Dynamic Discovery Option uses *well-known Oracle Names server*, which are precise names hard-coded into DNS or the HOSTS file on both the Oracle Names Server and its clients:

The well-known host names for TCP connections	The well-known computer Names for Named Pipes connections¹	The well-known service name for an SPX connection
oramesrvr0	ORANAMESRVR0	oramesrvr
oramesrvr1	ORANAMESRVR1	
oramesrvr2	ORANAMESRVR2	
oramesrvr3	ORANAMESRVR3	
oramesrvr4	ORANAMESRVR4	

¹ Well-known Names Server names for Names Pipes must be in all uppercase.

Oracle Names servers then become available at these well known addresses, so that clients did not need to be told, by way of preferred Oracle Names server lists, where to find an Oracle Names server.

If the Dynamic Discovery Option was chosen, each Oracle Names server automatically replicated its data to all other well-known Oracle Names server in the region. Listeners were configured to register themselves with well-known Oracle Names servers.

See the *Oracle Names Administrator's Guide*, Release 2.0, for configuration information.

Oracle Names version 8

In Oracle Names version 8, the administrator may choose between continuing Oracle Names Server configuration as in version 1 or version 2, or using the new functionality. Oracle Names version 8 incorporates version 2 Dynamic Discover Option features without the constraints of a single region and single domain. The main features of Oracle Names version 8 include:

- A service can register itself with any Oracle Names server it can find, and its name and address are made available to all Oracle Names server in the region. Similarly, if an administrator manually registers a service to any Oracle Names server, that service information is available to all other Oracle Names Servers. The address information is shared in one of two ways:
 - Service Replication — In service replication, service information is stored in an Oracle Names server's cache and is instantly replicated to the caches of all other Oracle Names servers.
 - Region Database — If an Oracle database is used as the registration repository, the registration information is stored in the database, and from there is accessible to all the Oracle Names Servers.
- A list of Oracle Names servers to contact is created on a client through a discovery process rather than manual configuration, as described in ["Understanding Discovery"](#) on page 2-45.
- A client-side process may be run to create a client cache version of the data stored in Oracle Names servers, including database addresses, Oracle Names server addresses, as well as other service information. As a result, the client does not have to contact an Oracle Names server for address information, reducing address look up time.
- Oracle Names version 8 does not require a database to hold topology information. However, an administrator may choose to use one, and it is recommended.

See ["Configuring the Network with the Oracle Names Method"](#) on page 6-21 for configuration information.

Understanding Discovery

A list of Oracle Names servers is created that allows a client or another Oracle Names server contact an Oracle Names server. This process of creating the list is called discovery.

When a client or an Oracle Names server, tries discover Oracle Names servers, it tries to find one Oracle Names server. Once the client finds an Oracle Names server, it pings all other Oracle Names server in the region. A list of Oracle Names servers is then created on the client and saved to .SDNS.ORA on UNIX and SDNS.ORA on Windows platforms. This list is sorted in order of response time.

Discovery searches for the first Oracle Names server in the following order:

1. a [preferred Oracle Names Server](#) in the SQLNET.ORA file
2. a [well-known Oracle Names server](#)
3. a local Oracle Names server configured with TCP/IP on port 1575

If the client is unable to find the first Oracle Names using the above methods:

- You will be prompted to specify the address of an Oracle Names server if you are using the Net8 Assistant.
- The NAMESCTL control utility will fail. You must re-run the command with the address specified.

Oracle Connection Manager Architecture

Oracle Connection Manager acts like a router through which client connection requests may either be sent to its next hop or directly to a server. Clients who route their connection requests through a Connection Manager may then take advantage of the connection concentration, Net8 access control, or multi-protocol support features configured on that Connection Manager.

Oracle Connection Manager Processes

There are three main processes associated with Oracle Connection Manager:

- [CMGW](#)
- [CMADMIN](#)
- [CMCTL](#)

CMGW

CMGW is a gateway process acting as a hub for the Oracle Connection Manager. This process is responsible for the following:

- registering with the [CMADMIN](#) process
- listening for incoming connection requests. By default it listens on TCP/IP with port 1630
- initiating connection requests to listeners for clients
- relaying data between the client and server
- answering requests initiated by CMCTL

CMADMIN

CMADMIN is a multi-threaded process that is responsible for all administrative issues of the Oracle Connection Manager. This process is responsible for the following:

- processing the CMGW registration
- identifying all listeners serving at least one database instance
- registering source route address information about the CMGW and listeners
- answering requests initiated by CMCTL
- monitoring registered listeners with the Oracle Names server, then source route addresses
- locating local Oracle Names servers
- maintaining address information in the Oracle Names server for the SQL*Net 2.x and Net 8.x clients

The Oracle Connection Manager periodically goes to the Oracle Names server to update its cache of available services.

CMCTL

CMCTL is the executable name for the Oracle Connection Manager control utility. It provides administrative access to CMADMIN and CMGW. For more information about the Oracle Connection Manager Control Utility, see "[Oracle Connection Manager Control Utility \(CMCTL\)](#)" on page A-89.

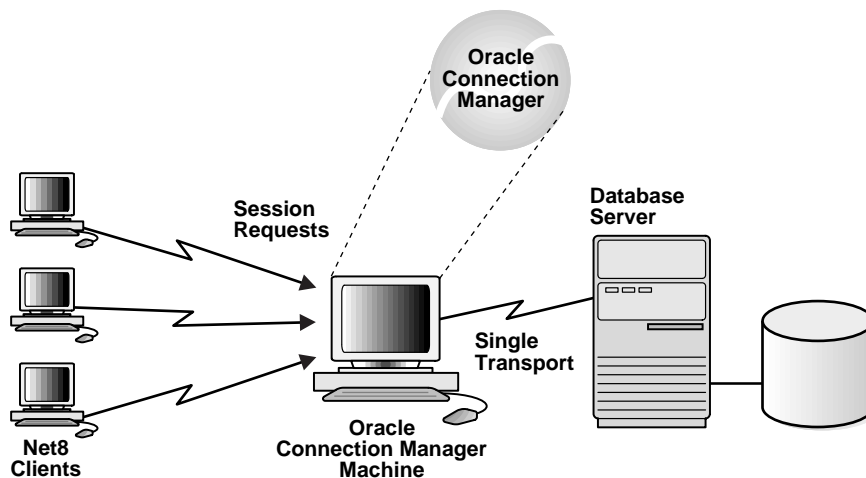
Connection Concentration

Oracle Connection Manager enables you to take advantage of Net8's ability to multiplex or funnel multiple client network sessions through a single transport protocol connection to a multi-threaded server destination.

Concentration reduces the demand on resources needed to maintain multiple connections between two processes by enabling the server to use fewer connection end points for incoming requests. This enables you to increase the total number of network sessions that a server can handle. By using multiple Connection Managers, it is possible for thousands of concurrent users to connect to a server.

Figure 2-16 depicts how concentration works:

Figure 2-16 Connection Concentration through Oracle Connection Manager



For configuration information, see "[Enabling Connection Concentration](#)" on page 7-31.

Net8 Access Control

Oracle Connection Manager also includes a feature which you can use to control client access to designated servers in a TCP/IP environment. By specifying certain filtering rules you may allow or restrict specific clients access to a server based on the following criteria:

- Source host name(s) or IP address(es) for clients
- Destination host name(s) or IP address(es) for servers
- Destination database service name

For configuration information, see "[Enabling Net8 Access Control](#)" on page 7-38.

Net8 Firewall Proxy

Some firewall vendors also offer the Net8 Firewall Proxy, which is installed on firewalls requiring an application proxy. The Net8 Firewall Proxy has the same functionality as the Oracle Connection Manager.

Note: Ask your firewall vendor if the Net8 Firewall Proxy is supported.

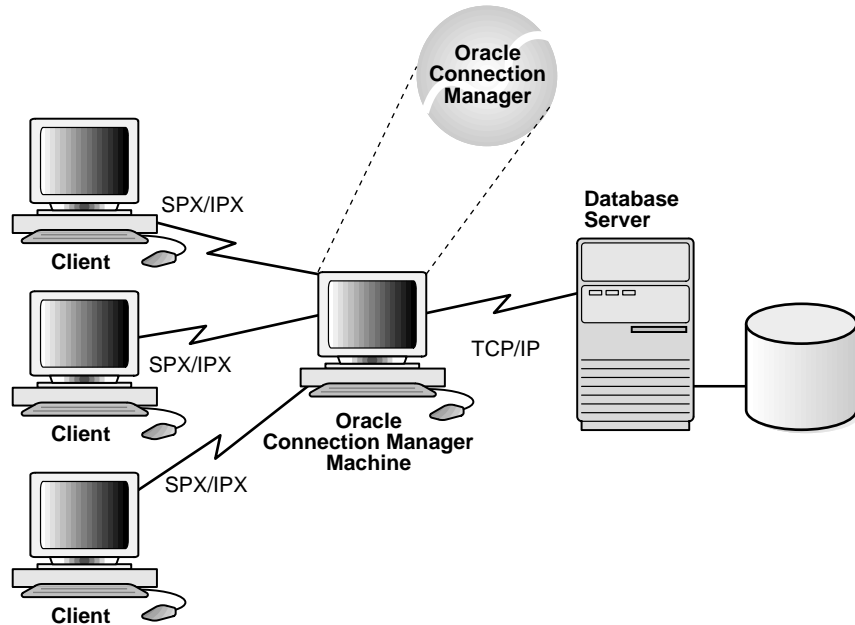
Multi-Protocol Support

Oracle Connection Manager also provides multiple protocol support enabling a client and server with different networking protocols to communicate with each other. This feature replaces functionality previously provided by the Oracle Multi-Protocol Interchange with SQL*Net version 2.

Net8 can traverse as many networking protocol stacks as can be installed and supported. In fact, the number of networking protocols supported is limited only by those restrictions imposed by the specific node's hardware, memory and operating system.

[Figure 2-17](#) depicts how a client in an SPX network can route its network session to a server over a TCP/IP transport through Oracle Connection Manager.

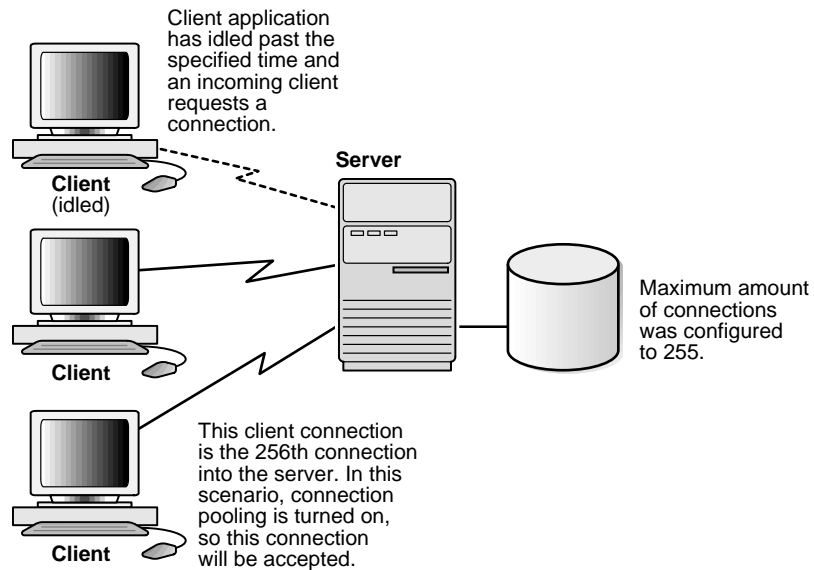
Figure 2–17 Multi-Protocol Support Through Oracle Connection Manager



For configuration information, see ["Enabling Multi-Protocol Support"](#) on page 7-35.

Connection Pooling

Connection pooling is a resource utilization and user scalability feature that allows you to maximize the number of physical network connections to a multi-threaded server. This is achieved by sharing or pooling a dispatcher's set of connections among multiple client processes. [Figure 2–18](#) shows how connection pooling works:

Figure 2–18 Connection Pooling

By using a time-out mechanism to temporarily release transport connections that have been idle for a specified period of time, connection pooling makes these physical connections available for incoming clients, while still maintaining a logical network session with the previous idle connection. When the idle client has more work to do, the physical connection is reestablished with the dispatcher.

For configuration information, see "[Enabling Connection Concentration](#)" on page 7-31.

Part II

Net8 Setup and Configuration

Part II describes how to setup, configure, connect, and troubleshoot Net8, as well as implement advanced features. Part II contains the following chapters:

- [Chapter 3, "Planning Your Network"](#)
- [Chapter 4, "Understanding Release 8.1 Installation and Net8 Products"](#)
- [Chapter 5, "Considerations for Upgrades and Migration"](#)
- [Chapter 6, "Configuring Naming Methods and the Listener"](#)
- [Chapter 7, "Enabling Optional Net8 Features"](#)
- [Chapter 8, "Establishing a Connection and Testing the Network"](#)
- [Chapter 9, "Configuring Multi-Threaded Server"](#)
- [Chapter 10, "Enabling Net8 Enhancements for Programmers"](#)
- [Chapter 11, "Troubleshooting Net8"](#)

Planning Your Network

Net8 provides a variety of options to help you design and manage networks that are both flexible and easy to use. With Net8's enhanced scalability and manageability features, you can develop a network to support a wide range of environments whether they be simple workgroups or large mission critical enterprises.

This chapter describes considerations for planning a network using Net8. It explains the relationships of the network products, and options for expanding and better managing your future network. It includes the following sections:

- [Planning Overview](#)
- [Defining Your Network Layout](#)
- [Resolving Net Service Names](#)
- [Improving Large Network Performance](#)
- [Planning Summary](#)

Planning Overview

Take the time to review and plan your network before you configure it. As you are planning your Oracle network, remember to keep future needs in mind as well as present requirements. Some of the more important decisions which you will need to make regarding your network include:

- [Defining Your Network Layout](#)
- [Resolving Net Service Names](#)
- [Improving Large Network Performance](#)

Defining Your Network Layout

The following checklist is provided to help you outline the main components of your network.

1. Define from the outset what it is you hope to accomplish with your network.
2. Research the functionality required by your client applications, then assess the resources that are available to meet those requirements.
3. Determine which machines or nodes are best suited for client or server applications.
4. Select a networking protocol which best suits your existing or future networking requirements. You may be able to choose a single transport level protocol that works well on all the components in your network. Protocols are available for most of the major protocols on many platforms. Your network may also involve clients or servers operating over more than one protocol.
5. If you decide to use multiple protocols on your network, determine which nodes are best suited to install Oracle Connection Manager. Your choice of nodes will be determined by the networking protocols you have chosen as well as the machine's capacity to handle anticipated traffic.

It helps sometimes to draw a picture of your network layout displaying the logical as well as physical relationships between networking components.

Resolving Net Service Names

Once you have defined your network layout, you will need to decide how best to configure and manage your network implementation. One of the first and most important decisions that you will need to make is choosing a naming method.

Naming Methods

Naming method refers to the method used by a client application to resolve a net service name to a network address when attempting to connect to a database service. Net8 provides four naming methods:

- [Host Naming](#)
- [Local Naming](#)
- [Centralized Naming Using Oracle Names](#)
- [External Naming](#)

Depending on the size and characteristics of your network, each method will have positive and negative implications for both how the network is configured and administered.

Host Naming

Host naming enables users in a TCP/IP environment to resolve net service names using their existing name resolution service. This name resolution service might be DNS, NIS, or simply a centrally-maintained set of HOSTS files.

Host naming allows users to connect to an Oracle server simply by using the a host name alias. The connection is established by using the default TCP/IP port for the listener, that is, port 1521. Multiple databases per node and database location transparency are supported through matching listener global database names with host name aliases.

Host naming can eliminate the need for a local naming configuration file (TNSNAMES.ORA) in environments where simple database connectivity is desired. It is not however, suitable for large, complex environments where advanced features such as connection pooling, Oracle Connection Manager, or heterogeneous services.

For configuration information, see "[Configuring the Network with the Host Naming Method](#)" on page 6-7.

Establishing a Connection Using the Host Naming Option

The process for establishing a client session using the host naming option is as follows:

1. The client initiates a connect request providing a TCP/IP host name alias.
2. Host naming resolves this host name or alias by generating a network address using the net service name as both the TCP/IP hostname and the global database name. The TCP/IP port defaults to 1521.
3. Net8 makes the connect request to the address created.
4. A listener, listening at registered TCP/IP port 1521, receives the request and establishes a connection to the database.
5. The connection is accepted by the server.

Local Naming

Local naming refers to the method of resolving a net service name to a network address by using information configured on each individual client. Much like an address book, this information is entered in a local naming configuration file called TNSNAMES.ORA.

For configuration information, see "[Configuring the Network with the Local Naming Method](#)" on page 6-12.

Establishing a Connection Using Local Naming

The process for establishing a client session using local naming is as follows:

1. The client initiates a connect request providing a net service name.
2. The net service name is resolved to a network address configured in a local naming file.
3. Net8 makes the connect request to the address provided.
4. A listener receives the request and directs it to the database it is servicing.
5. The connection is accepted by the server.

Centralized Naming Using Oracle Names

Centralized Naming refers to the method of resolving a net service name to a network address by using Oracle Names. Oracle Names uses Oracle Names servers to store the names and addresses of all database services on a network. Much like people calling for directory assistance, clients wishing to connect to a server direct their connect requests to an Oracle Names server. Oracle Names servers resolve the net service name to a network address and return that information to the client.

For configuration information, see "[Configuring the Network with the Local Naming Method](#)" on page 6-12.

Establishing a Connection Using the Centralized Naming Option

The process for establishing a client session using the centralized naming option is as follows:

1. The client initiates a connect request providing a net service name.
2. The connect request is forwarded to an Oracle Names server where the net service name is resolved to a network address. This address is returned to the client.

3. Net8 makes the connect request to the address provided.
4. A listener receives the request and redirects it to the database it is servicing.
5. The connection is accepted by the server.

External Naming

External naming refers to the method of resolving a net service name to a network address by using a supported non-Oracle naming service. External naming resolves net service names stored in customers' external (non-Oracle) naming services. They include:

- Network Information Service (NIS) External Naming
- NetWare Directory Service (NDS) External Naming

Note: In previous releases of SQL*Net, these external naming methods were part of the Oracle Advanced Networking Option. They are now included as a standard part of Net8.

Distributed Computing Environment Cell Directory Service (CDS) continues to be available as part of the DCE Integration part of Oracle Advanced Networking Option.

Establishing a Connection Using External Naming

The process for establishing a client session using external naming is as follows:

1. The client initiates a connect request providing a net service name.
2. An external naming method forwards the request to an external naming system that resolves the net service name to a network address. The address is returned to the client.
3. Net8 makes the connect request to the address provided.
4. A listener receives the request and redirects it to the database it is servicing.
5. The connection is accepted by the server.

Configuring External Naming

To configure external naming using the Net8 Assistant, proceed as follows:

1. Verify that the applicable external naming method has been installed on the client node.
2. Specify the use of an external naming method (for example, CDS, NDS or NIS) in your profile. If it is not, use the Net8 Assistant to edit the client profile.

For configuration information, see "[Configuring the Network with the External Naming Method](#)" on page 6-51.

Oracle Names and External Naming

Oracle Names can be used in conjunction with other proprietary or open naming services to provide cross-environment name resolution. For example, external naming for CDS/DCE, NIS or NDS could be installed on all clients and servers in an enterprise network already running Oracle Names to provide name resolution across multiple name services.

Since Oracle Names is a proprietary name service storing and resolving names and addresses for Oracle databases only, one names solution could be to store all your Oracle services in Oracle Names, and use a directory service such as DNS or X.500 as your global naming service.

Choosing a Naming Method

The table below summarizes the relative advantages and disadvantages of each naming method and provides recommendations for using them in your network.

Naming Method	Advantages/Disadvantages	Recommended for:
Host Naming	<ul style="list-style-type: none">■ Requires minimal user configuration. The user may provide only the name of the host to establish a connection.■ Eliminates the need to create and maintain a local names configuration file (TNSNAMES.ORA).■ Eliminates the need to understand Oracle Names administration procedures. <p>Disadvantages: Available only if all of the following are true:</p> <ul style="list-style-type: none">■ Your client and server are connecting using TCP/IP.■ The host name is resolved through an IP address translation mechanism such as Domain Name Services (DNS), Network Information Services (NIS), or a centrally maintained TCP/IP hosts file.■ No Oracle Connection Manager features are requested.	Simple TCP/IP networks that meet the criteria listed.
Local Naming	<ul style="list-style-type: none">■ Provides a relatively straightforward method for resolving net service name addresses.■ Resolves net service names across networks running different protocols. <p>Disadvantages: Requires local configuration of all net service name and address changes.</p>	Simple distributed networks with a small number of services that change infrequently.

Naming Method	Advantages/Disadvantages	Recommended for:
Oracle Names	<ul style="list-style-type: none">■ Centralizes network names and addresses in a single place, facilitating administration of name changes and updates. For example, whenever a change is made to an existing server or a new server is added to the network, the change is made only once on one Oracle Names server. This eliminates the need for an administrator to make changes to what potentially could be hundreds or even thousands of clients.■ Resolves net service names across networks running different protocols <p>Disadvantages:</p> <ul style="list-style-type: none">■ Oracle Names stores network names and addresses for Oracle services only.■ Requires additional setup and administration of Oracle Names servers.	Large, complex networks (over 20 databases) that change on a frequent basis.
External Naming	Allows administrators to load Oracle net service name into their native name service using tools and utilities with which they are already familiar.	Networks with existing name services.

Improving Large Network Performance

You may improve the performance of large networks by implementing one of the following:

- [Managing Connection Requests](#)
- [Connection Pooling](#)
- [Connection Concentration](#)
- [Client Load Balancing](#)
- [Optimizing Data Transfer by Adjusting the Session Data Unit \(SDU\) Size](#)
- [Persistent Buffer Flushing for TCP/IP](#)
- [Configuring Listener Queuesize](#)

Managing Connection Requests

If you expect your network to receive excessive connection traffic, you can use the listener to manage these requests by redirecting them to either prestarted or prespawned dedicated servers or shared servers.

The table below summarizes the relative advantages of each, and provides recommendations for using them in your network.

Type	Advantages	Recommended for:
Shared servers	<ul style="list-style-type: none">■ Utilizes network resources more efficiently than a dedicated server, thus increasing the throughput and performance of your sessions.■ Enables you to minimize the memory and processing resources needed on the server side as the number of sessions to the database increases.	Networks where the Oracle shared or multi-threaded server is supported, or where the creation of a new server is slow and resource-intensive.
Prestarted or prespawned dedicated servers (the default)	<ul style="list-style-type: none">■ Reduces connect time by eliminating the need to create a dedicated server for each new connection request.■ Provides better use of allocated memory and system resources by recycling servers for use by other connections without having to shut down and recreate a server.	Networks where the Oracle shared or multi-threaded server is not supported, or where the creation of a new server is slow and resource-intensive.

For more information about configuring your listener to redirect connect requests to either prestarted or prespawnd dedicated servers, see [Chapter 9, "Configuring Multi-Threaded Server"](#).

Connection Pooling

Connection pooling is a resource utilization feature that allows you to maximize the number of physical network connections to a multi-threaded server. This is achieved by sharing or pooling a dispatcher's set of connections among multiple client processes.

By using a time-out mechanism to temporarily release transport connections that have been idle for a specified period of time, connection pooling makes these physical connections available for incoming clients, while still maintaining a logical session with the previous idle connection. When the idle client has more work to do, the physical connection is reestablished with the dispatcher.

This feature only works when multi-threaded server is configured.

Connection pooling concepts are described in "[Connection Pooling](#)" on page 2-50, and configuration information is described in "[Enabling Connection Pooling](#)" on page 7-49.

Connection Concentration

Connection concentration is a feature that is available through Oracle Connection Manager. It allows you take advantage of Net8's ability to multiplex or funnel multiple client sessions over a single transport to a multi-threaded server. Like connection pooling, concentration optimizes network resources and increases the number of client-server sessions that are possible across a fixed number of physical server ports. Unlike connection pooling, concentration maintains the transport connection.

Connection concentrations concepts are described in "[Oracle Connection Manager Architecture](#)" on page 2-46, and configuration information is described in "[Enabling Connection Concentration](#)" on page 7-31.

Using Connection Pooling and Concentration

The table below summarizes the relative advantages with using connection pooling and concentration and provides recommendations for using them in your network.

Feature	Advantages	Recommended for:
Connection Pooling	<ul style="list-style-type: none"> ▪ Limits the number of network resources used per process. ▪ Maximizes the number of client/server sessions over a limited number of physical connections. ▪ Optimizes resource utilization. 	Networks where many clients run interactive "high think/search time" applications such as messaging and OLAP.
Concentration	<ul style="list-style-type: none"> ▪ Supports large client populations. ▪ Allows identification and monitoring of real users. ▪ Allows mid-tier applications to support additional services. ▪ Requires only a single transport for clients with multiple applications. ▪ Requires only a single network connection for database links. 	Networks where "continuous" connectivity is required.

Client Load Balancing

Load balancing is a feature by which client connections are distributed evenly among multiple listeners, dispatchers, instances, and nodes so that no single component is overloaded. By balancing the load, you can improve connection performance.

You can configure a client to randomly chooses between listeners for its connect requests. When client load balancing is set to on, Net8 balances the load over the list of listener addresses by picking one at random. When set to off, Net8 uses the elements in the list in the order in which they are provided. For configuration information, see "[Configuring Multiple Address Options](#)" on page 7-11.

Optimizing Data Transfer by Adjusting the Session Data Unit (SDU) Size

Tuning your application to reduce the number of round trips across the network is the best way to improve your network performance. If this is done, it is also possible to optimize data transfer by adjusting the size of the session data unit (SDU).

The SDU is a buffer that Net8 uses to place data before transmitting across the network. Net8 sends the data in the buffer either when requested or when it is full.

The table outlines considerations when modifying the size of the SDU may or may not be appropriate.

Modify session data unit size when:	Do not modify session data unit size when:
<ul style="list-style-type: none"> ■ The data coming back from the server is fragmented into separate packets ■ You are on a wide area network (WAN) that has long delays ■ Your packet size is consistently the same ■ Large amounts of data are returned 	<ul style="list-style-type: none"> ■ Your application can be tuned to account for the delays ■ You have a higher speed network where the effect of the data transmission is negligible ■ Your requests return small amounts of data from the server

Note: The SDU size should be set as a multiple of the normal transport frame size. Since the normal Ethernet frame size is 1024, the most efficient SDU size over an Ethernet protocol should be a multiple of 1024, but not more than four times the amount of 1024.

If you are using either connection pooling or connection concentration, keep in mind that these features require an additional 16 bytes per transport. For more information about the protocol frame size, refer to your protocol specific documentation.

You may adjust the session data unit size by adding a parameter in your local naming configuration file (TNSNAMES.ORA). For more information, refer to "[Configuring Advanced Connection Attributes for a Net Service Name](#)" on page 7-7.

Persistent Buffer Flushing for TCP/IP

Under certain conditions in some applications using TCP/IP, Net8 packets may not get flushed immediately to the network. Most often, this behavior occurs when large amounts of data are streamed from one end to another. The implementation of TCP/IP itself is the reason for the lack of flushing, and can cause unacceptable delays. To remedy this problem, you can specify no delays in the buffer flushing process.

For more information, see the TCP.NODELAY parameter in "[Protocol-Specific Parameters \(PROTOCOL.ORA\)](#)" on page C-81.

Configuring Listener Queuesize

If you anticipate receiving a large number of connection requests for a listening process (such as a listener, Oracle Connection Manager or Oracle Names) over TCP/IP, Net8 allows you to configure the listening queue to be higher than the system default.

For more information, see "[Handling Larger Volumes of Connection Requests](#)" on page 6-70.

Planning Summary

The table below summarizes many of the options you may have chosen as you planned your network.

Subject	Options
Network Layout	<ul style="list-style-type: none">▪ Single or Multiple Protocols
Net Service Name Resolution	<ul style="list-style-type: none">▪ Host Naming▪ External Naming▪ Centralized Naming▪ Local Naming
Connection Request Management	<ul style="list-style-type: none">▪ Dedicated Servers▪ Prestarted/Prespawned Dedicated Servers▪ Shared Servers
Network Performance	<ul style="list-style-type: none">▪ Connection Pooling▪ Connection Concentration▪ Client Load Balancing▪ Optimizing the Session Data Unit Size▪ Persistent Buffer Flushing▪ Increasing the Listener Queue Size

Understanding Release 8.1 Installation and Net8 Products

This chapter describes release 8.1 configuration files, installation and Net8 products. Specific topics discussed are:

- [New Products and Features In This Release](#)
- [Net8 Release 8.1 Configuration File Changes](#)
- [Net8 Installation and Net8 Products](#)

New Products and Features In This Release

The following table describes the *new* products and features of Net8 release 8.1.5:

New Product/New Feature	Description	For further information, see...
Database Instance Registration	<p>Database instances register themselves with the listener when started. Prior to this release, information about the instance had to be manually configured in the LISTENER.ORA file.</p> <p>Instance registration is comprised of the following:</p> <ul style="list-style-type: none">▪ service registration, which provides the listener with instance information, such as database service names and instance names.▪ MTS dispatcher registration, which provides dispatcher information to the listener <p>Database instance registration also permits automatic connect-time failover of a client connect request to a different listener if an instance is down.</p>	"Database Instance Registration to Listeners" on page 2-27
Connection Load Balancing	<p>Instance registration enables connection load balancing. Connection load balancing balances the number of active connections among various instances and dispatchers for the same service. This enables listeners to make their routing decisions based on how many connections each dispatcher has <i>and</i> on how loaded the nodes that the instances run.</p>	"Connection Load Balancing" on page 2-30

New Product/New Feature	Description	For further information, see...
Service Naming	<p>Service naming can include multiple services provided by a single database and services that span multiple instances.</p> <p>Up to this release, the client was configured with the Oracle System Identifier (SID) of a database instance. This SID was then passed to the listener. The listener would then verify this information and permit or deny a connection. The SID was also used internally by the database as pointer to the System Global Area (SGA).</p> <p>While a SID identified a database instance, it did not identify a database. This limitation caused a database to have no more than one service associated with it.</p> <p>Because a database can serve multiple services, SID has been replaced with service naming. Service naming allows clients to access:</p> <ul style="list-style-type: none"> ▪ a service as a whole through the service name ▪ an instance of a database (the way SID did) through the instance name <p>To support services that include multiple instances, use the following new parameters in connect descriptors:</p> <ul style="list-style-type: none"> ▪ SERVICE_NAME ▪ INSTANCE_NAME 	<ul style="list-style-type: none"> ▪ "Connect Operations" on page 2-2 ▪ "Net8 Release 8.1 Configuration File Changes" on page 4-8 ▪ "Setting Service Names and Instance Names" on page 6-3

New Product/New Feature	Description	For further information, see...
<p><i>net_service_name</i> replaces <i>service_name</i></p>	<p>In previous releases, an alias for a connect descriptor (listener address and service information) was called a <i>service_name</i>. For example:</p> <pre> service_name= (description= (address= (protocol_address_information)) (connect_data= (sid=sid))) </pre> <p>End users were instructed to connect to a service with it. For example:</p> <pre> sql> CONNECT username/password@service_name </pre> <p>Because service name now refers to a service in the connect descriptor, <i>service_name</i> has been replaced with <i>net_service_name</i> for the connect descriptor and in the connect string. A <i>net_service_name</i> mapped to a connect descriptor looks like:</p> <pre> net_service_name= (description= (address=(protocol specific address data)) (connect_data= (service_name=service_name))) </pre> <p>An end user identifies a service with <i>net_service_name</i> in a connect string:</p> <pre> sql> connect username/password@net_service_name </pre>	<p>Chapter 6, "Configuring Naming Methods and the Listener"</p>

New Product/New Feature	Description	For further information, see...
DB_DOMAIN	<p>DB_DOMAIN in the initialization file (INITSID.ORA) no longer has a default setting of .WORLD. The new setting is NULL. Therefore, service names do not need to include the domain, if the default setting is used. In prior releases of Oracle, the default setting was .WORLD.</p> <p>If DB_DOMAIN is set to NULL, the service name does not need to be domain-qualified.</p> <p>If the DB_DOMAIN is <i>not</i> set to NULL, ensure the service name you enter includes the domain. For example, if an Oracle8i database has a service name of SALES and a domain of COM, you specify a service name of SALES.COM to identify the service.</p>	"Setting Service Names and Instance Names" on page 6-3
Java Option Support	<p>Prior to this release, Net8 only supported connections that used the Two-Task Common (TTC) presentation layer and Transparent Network Substrate (TNS) Network session (NS) layer to establish client connections.</p> <p>The Java option allows customers to program the database server using traditional database stored procedures, Enterprise JavaBeans and CORBA Servers. To support clients accessing Enterprise JavaBeans and CORBA Servers in the database, the presentation layer and session layer support have been expanded.</p>	"Configuring Java Option Connections" on page 7-50
Net8 Configuration Assistant	<p>The Net8 Configuration Assistant is post-installation tool that performs basic configuration. After installation, it automatically configures default configuration files.</p> <p>In addition, the Net8 Configuration Assistant may be run in stand-alone mode to configure various elements of configuration, including the:</p> <ul style="list-style-type: none"> ■ listener ■ naming methods ■ net service names ■ LDAP directory service. 	"Net8 Installation and Net8 Products" on page 4-21
RADIUS (Remote Authentication Dial-In User Service) authentication method	<p>RADIUS is a lightweight, open protocol for user authentication, authorization, and accounting between a network client and an authentication server. The RADIUS authentication method enables authentication with RADIUS-compliant devices such as token cards. In addition, RADIUS supports challenge-response and accounting.</p>	<i>Oracle Advanced Security Administrator's Guide</i>

New Product/New Feature	Description	For further information, see...
TCP/IP with SSL Protocol	<p>The TCP/IP with SSL protocol enables client/server conversation over a network using TCP/IP and the Secure Sockets Layer (SSL).</p> <p>SSL stores authentication data, such as certificates and private keys, in an Oracle Wallet. When the client initiates a Net8 connection to the server, SSL performs a handshake between the two using the private key and certificate. During the handshake the following processes occur:</p> <ul style="list-style-type: none"> ■ The client and server negotiate a cipher suite, a set of authentication, encryption, and data integrity types, to apply to the messages they exchange. ■ Depending on its configuration, the server may then send its own certificate to the client in a message encrypted with the client's public key. The server may also send a request for the client's certificate in the same message. The client decrypts this message by using its own private key, then verifies that the server's certificate bears the certificate authority's signature. ■ If required, the client may send the user's certificate to the Oracle server. The certificate ensures that the user's information is correct and that the public key actually belongs to that user. <p>The server checks the user's certificate to verify that it bears the certificate authority's signature.</p>	<i>Oracle Advanced Security Administrator's Guide</i>

New Product/New Feature	Description	For further information, see...
Secure Sockets Layer (SSL) authentication method	<p>Stores authentication data, such as certificates and private keys, in an Oracle Wallet. When the client initiates a Net8 connection to the server, SSL performs a handshake between the two using the private key and certificate. During the handshake the following processes occur:</p> <ul style="list-style-type: none">■ The client and server negotiate a cipher suite—a set of authentication, encryption, and data integrity types—to apply to the messages they exchange.■ Depending on how you have configured SSL, the client may send the user's certificate to the Oracle server. The certificate ensures that the user's information is correct and that the public key actually belongs to that user. The server checks the user's certificate to verify that it bears the certificate authority's signature.■ Again, depending on your configuration, the server may then send its own certificate to the client in a message encrypted with the client's public key. The client decrypts this message by using its own private key, then verifies that the server's certificate bears the certificate authority's signature.	<i>Oracle Advanced Security Administrator's Guide</i>
TCP/IP with SSL Protocol	Enables client/server conversation over a network using TCP/IP and SSL.	<i>Oracle Advanced Security Administrator's Guide</i>

Net8 Release 8.1 Configuration File Changes

This section describes the following:

- [Configuration File Syntax Changes](#)
- [TNSNAMES.ORA Changes](#)
- [LISTENER.ORA Changes](#)
- [Obsolete Parameters for Release 8.1](#)

Configuration File Syntax Changes

The table below describes the syntax differences between Net8 release 8.0 and Net8 release 8.1.

Effectuated element	Net8 release 8.0	Net8 release 8.1
Connecting with service name and instance name rather than system identifier (SID) in TNSNAMES.ORA	<p>The SID of the database had to be specified in the CONNECT_DATA section of the TNSNAMES.ORA file, as shown below:</p> <pre> service_name= (description= (address_list= (address=...) (address=...)) (connect_data=(sid=sales)) </pre>	<p>Because a database can include services that span multiple instances, SID has been replaced by service name and, optionally, instance name.</p> <p>The following entry allows a client to connect to the SALES.COM service:</p> <pre> net_service_name= (description= (address=...) (address=...) (connect_data= (service_name=sales.com))) </pre> <p>The following entry allows a client to connect to the OP1 instance, which is a part of the OP.COM service:</p> <pre> net_service_name= (description= (address=...) (address=...) (connect_data= (service_name=op.com) (instance_name=op1))) </pre> <p>SERVICE_NAME is typically the global database name, a name comprised of the database name and domain name, entered during installation or database creation. INSTANCE_NAME is typically the SID entered during installation or database creation.</p> <p>Additional Information: See:</p> <ul style="list-style-type: none"> ▪ "Setting Service Names and Instance Names" on page 6-3 ▪ "Configuring the Network with the Local Naming Method" on page 6-12

Affected element	Net8 release 8.0	Net8 release 8.1
<p>Client load balancing with DESCRIPTION_LIST and ADDRESS_LIST in TNSNAMES.ORA</p>	<p>DESCRIPTION_LISTs were used for mapping multiple connect descriptors to net service names and for client load balancing of multiple listeners.</p> <p>In order to client load balance, the user had to define separate connect descriptors for each listener.</p>	<p>While client load balancing is still ON by default for DESCRIPTION_LISTs, load balancing can also be explicitly specified with the new LOAD_BALANCE parameter for an ADDRESS_LIST or associated with a set of ADDRESSes or set DESCRIPTIONs. The following entry associates client load balancing with a list of addresses, affecting all listener ADDRESSes:</p> <pre data-bbox="822 522 1179 748"> net_service_name= (description= (load_balance=on) (address=...) (address=...) (connect_data= (service_name=sales.com))) </pre> <p>Client load balancing works with any version of the client, but requires an Oracle8i release 8.1 server.</p> <p>Additional Information: See:</p> <ul style="list-style-type: none"> ▪ "Configuring Multiple Address Options" on page 7-11 ▪ Figure C-4, "Client Load Balancing for Release 8.1 and Prior to Release 8.1" on page C-35.

Effected element	Net8 release 8.0	Net8 release 8.1
<p>Nested addresses within ADDRESS_LIST in TNSNAMES.ORA and LISTENER.ORA</p>	<p>If there was only one list of addresses, it was required to use an ADDRESS_LIST, as shown below:</p> <pre> service_name= (description= (address_list= (address=...) (address=...)) (connect_data=(sid=sales)) </pre>	<p>Address lists in Net8 do not have to be embedded in an ADDRESS_LIST if there is only one list, as shown below:</p> <pre> net_service_name= (description= (address=...) (address=...) (connect_data= (service_name=sales.com))) </pre> <p>Nested addresses work with any version of the client, but requires an Oracle8i release 8.1 server.</p> <p>Additional Information: See:</p> <ul style="list-style-type: none"> ▪ "Configuring the Network with the Local Naming Method" on page 6-12 ▪ "Local Naming Parameters (TNSNAMES.ORA)" on page C-31

Effected element	Net8 release 8.0	Net8 release 8.1
Connect-Time Failover with ADDRESS_LIST in TNSNAMES.ORA	An ADDRESS_LIST implied connect-time failover, where a client connect fails over to a different listener if the first listener fails. Net8 and SQL*Net proceeded through all the listener addresses until one succeeded.	<p>Failover is now supported by default for ADDRESS_LISTs, DESCRIPTION_LISTs, and a set of DESCRIPTIONs. Failover can also be explicitly specified with the new FAILOVER parameter for a set of ADDRESSES. The following entry associates connect-time failover with a list of addresses, affecting all listener ADDRESSES:</p> <pre>net_service_name= (description= (failover=on) (address=...) (address=...) (connect_data= (service_name=sales.com)))</pre> <p>Connect-time failover works with any version of the client, but requires an Oracle8i release 8.1 server.</p> <p>Additional Information: See:</p> <ul style="list-style-type: none"> ▪ "Configuring Multiple Address Options" on page 7-11 ▪ "Local Naming Parameters (TNSNAMES.ORA)" on page C-31
Routing connections through an Oracle Connection Manager with SOURCE_ROUTE=ON in TNSNAMES.ORA	The SOURCE_ROUTE parameter, which creates a source route of addresses through an Oracle Connection Manager to the destination service, had to be outside of an ADDRESS_LIST:	<pre>service_name= (description= (source_route=on) (address_list= (address=...) (address=...)) (connect_data= (sid=sales)))</pre> <p>SOURCE_ROUTE is no longer outside of an ADDRESS_LIST. Instead, it is now associated with a list of ADDRESSES, as shown below:</p> <pre>net_service_name= (description= (source_route=on) (address=...) (address=...) (connect_data= (service_name=sales.com)))</pre> <p>Additional Information: See:</p> <ul style="list-style-type: none"> ▪ "Configuring the Client for Oracle Connection Manager Features" on page 7-41 ▪ "Local Naming Parameters (TNSNAMES.ORA)" on page C-31

Effected element	Net8 release 8.0	Net8 release 8.1
SID_LIST_listener_name information in the LISTENER.ORA	<p>The LISTENER.ORA required a definition of the SID of the database served by the listener. The SID of the database had to be specified in the SID_LIST section of the LISTENER.ORA file. For example:</p> <pre>sid_list_listener=(sid_list= (sid_desc= (global_dbname=sales.com) (sid_name=db1) (oracle_home=/usr/bin/oracle)))</pre> <p>Additional Information: See "Configuring the Listener" on page 6-62.</p>	<p>Because database instances and multi-threaded server dispatchers now register themselves with the listener, it is no longer necessary to explicitly define database information, unless you:</p> <ul style="list-style-type: none"> are connecting to an Oracle 8 release 8.0 or Oracle7 database: using a management tool, such as Oracle Enterprise Manager, that requires this information. <p>Additional Information: See "Configuring the Listener" on page 6-62.</p>
Presentation layer and Session layer information in LISTENER.ORA	<p>The only type of clients supported by Net8 were those connecting to the database using the Two-Task Common (TTC) presentation layer and the Transparent Network Substrate (TNS) Network Session (NS) session layer. This information was not required in the LISTENER.ORA file.</p>	<p>Besides traditional Net8 clients, this release also supports IIOP clients connecting to the Java option in the database. These clients use General Inter-Orb Protocol (GIOP) as the presentation layer rather than Two-Task Common (TTC). In addition, no session layering is needed.</p> <p>TTC and NS are the default and do <i>not</i> need to be explicitly defined in LISTENER.ORA for typical Net8 clients. If you have IIOP clients, the presentation and session layers must be explicitly defined. For example:</p> <pre>listener= (description= (protocol_stack= (presentation=giop) (session=raw)) (address= (protocol=tcp) (host=sales-pc) (port=2481)))</pre> <p>Additional Information: See "Configuring Java Option Connections" on page 7-50.</p>

Effected element	Net8 release 8.0	Net8 release 8.1
Address for CMADMIN process in CMAN.ORA	An address was only required for the Oracle Connection Manager gateway process, CMGW. The administrative process, CMADMIN used an internal IPC address.	<p>A protocol address is required for both CMADMIN and CMGW.</p> <p>This new address is depicted by the new CMAN_ADMIN parameter, as shown below:</p> <pre> cman=(address=(protocol=tcp)(host=anyhost)(port=1630)) cman_ admin=(address=(protocol=tcp)(host= cman-pc)(port=1650)) </pre> <p>By default the CMADMIN process listens on TCP/IP with a registered port of 1830.</p> <p>Additional Information: See:</p> <ul style="list-style-type: none"> ■ "Oracle Connection Manager Processes" on page 2-46 for conceptual information ■ "Enabling Connection Concentration" on page 31 for configuration information

TNSNAMES.ORA Changes

The table below describes the new parameters in the TNSNAMES.ORA file for Net8 release 8.1.

Parameter	Description
FAILOVER	<p>When set to ON, instructs Net8 at connect time to fail over to a different listener if the first listener fails. It determines how many addresses are tried, as shown below:</p> <pre> net_service_name= (description= (failover=on) (address=...) (address=...) (connect_data= (service_name=service_name))) </pre> <p>When set to OFF, instructs Net8 to try one address.</p> <p>By default, this parameter is set to ON for ADDRESS_LISTs, DESCRIPTION_LISTs and a set of DESCRIPTIONs.</p> <p>Additional Information: See:</p> <ul style="list-style-type: none"> ▪ "Configuring Multiple Address Options" on page 7-11 ▪ "Local Naming Parameters (TNSNAMES.ORA)" on page C-31

Parameter	Description
INSTANCE_NAME	<p>In addition to using the SERVICE_NAME parameter, the INSTANCE_NAME parameter can also be used to identify the database instance to access. INSTANCE_NAME is only necessary for an Oracle Parallel server database, where the specific instance to which to connect is important to identify. For example, shown below are descriptions for a service, OP, that spans multiple instances, OP1 and OP2. The end user can connect to the OP service through either instance.</p> <pre>net_service_name= (description= (address= (protocol=tcp) (host=opsnt1) (port=1521)) (connect_data= (service_name=op.com) (instance_name=op1)) net_service_name= (description= (address= (protocol=tcp) (host=opsnt1) (port=1521)) (connect_data= (service_name=op.com) (instance_name=op2))</pre> <p>The INSTANCE_NAME is typically the SID entered during installation or database creation.</p> <p>Additional Information: See "Configuring Multiple Address Options" on page 7-11 for information about the instance name string.</p>

Parameter	Description
LOAD_BALANCE	<p>When set to ON, instructs Net8 to progress through the list of listener addresses in a random sequence, balancing the load on the various listeners, as shown below:</p> <pre data-bbox="505 366 862 591"> net_service_name= (description= (load_balance=on) (address=...) (address=...) (connect_data= (service_name=sales.com))) </pre> <p>When set to OFF, instructs Net8 to try the addresses sequentially until one succeeds.</p> <p>By default, this parameter is set to ON for DESCRIPTION_LISTS.</p> <p>Additional Information: See:</p> <ul style="list-style-type: none"> ■ "Configuring Multiple Address Options" on page 7-11 ■ "Local Naming Parameters (TNSNAMES.ORA)" on page C-31
SERVICE_NAME	<p>Identifies the service to which to connect, as shown below:</p> <pre data-bbox="505 881 891 1078"> net_service_name= (description= (address=...) (address=...) (connect_data= (service_name=sales.com))) </pre> <p>The SERVICE_NAME is typically set to the global database name, a name comprised of the database name and domain name, entered during installation or database creation.</p> <p>Additional Information: See "Setting Service Names and Instance Names" on page 6-3 for information about the service name string.</p>

LISTENER.ORA Changes

The table below describes the new parameters in the LISTENER.ORA file for Net8 release 8.1.

Parameter	Description
PROTOCOL_STACK	<p>Identifies the presentation and session layer information for a connection:</p> <pre>listener= (description_list= (description= (protocol_stack= (presentation=giop) (session=raw)) (address_list= (address=...))))</pre> <p>It is <i>not</i> necessary to use PROTOCOL_STACK if you are using the default presentation layer of Two-Task Common (TTC) and the default session layer of Network Session (NS).</p> <p>Additional Information: See "Stack Communications in a Typical Oracle Networking Environment" on page 2-9 for a description of the supported presentation and session layers.</p>
PRESENTATION	<p>Identifies the presentation layer:</p> <ul style="list-style-type: none"> ■ Typical Net8 clients use a presentation layer of TTC. ■ Java clients require a presentation layer of General Inter-Orb Protocol (GIOP).
SESSION	<p>Identifies the session layer:</p> <ul style="list-style-type: none"> ■ Typical Net8 clients use a session layer of NS. ■ Java clients require a presentation layer of RAW for no session layer.

CMAN.ORA Changes

The CMAN.ORA file has a new parameter to define the CMADMIN address:

Parameter	Description
CMAN_ADMIN	<p>Identifies the address for the administrative process CMADMIN process, as shown below:</p> <pre>cman_admin= (address= (protocol=tcp) (host= cman-pc) (port=1650))</pre> <p>When the Oracle Connection Manager Control Utility, CMCTL, is run, it locates the CMADMIN protocol address to execute its commands.</p> <p>It is <i>not</i> necessary to explicitly specify CMAN_ADMIN if you are using the default address of TCP/IP on port 1830.</p> <p>Additional Information: See:</p> <ul style="list-style-type: none">▪ "Enabling Connection Concentration" on page 7-31▪ "Oracle Connection Manager Control Utility (CMCTL)" on page A-89

Obsolete Parameters for Release 8.1

The table below describes the obsolete parameters for release 8.1.

Parameter	File	Description
AUTOMATIC_IPC	SQLNET.ORA	This parameter used to force sessions through IPC addresses. Due to performance issues, this parameter has been removed. You should configure and IPC address instead.
NAMES.USE_PLUG_AND_PLAY	NAMES.ORA	This was used to enable/disable the Dynamic Discovery Option. There are other mechanism available to discover other Oracle Names server. Additional Information: See " Configuring the Network with the Oracle Names Method " on page 6-21.
NAMES.DOMAIN_CHECKPOINT_FILE	NAMES.ORA	Name of the file used to checkpoint topology data (domains addresses of servers in the local region). You should use the NAMES.REGION_CHECKPOINT_FILE parameter to create a checkpoint file with this information.
NAMES.TOPOLOGY_CHECKPOINT_FILE	NAMES.ORA	Name of the file used to checkpoint domain data (all the database addresses and other data in the region). You should use the NAMES.REGION_CHECKPOINT_FILE parameter to create a checkpoint file with this information.

Net8 Installation and Net8 Products

When you run Oracle Universal Installer, you can choose to install products from a number of installation options. Each installation option has a subset of installation types. If you choose Custom, you can selectively install products. The following table describes what gets installed when you choose a particular installation type.

Installation Type	Description
Typical	Installs a preconfigured starter database, licensable Oracle options and cartridges, networking services, Oracle8i utilities, and online documentation. This type of installation is recommended for users who want the complete database package.
Minimal	Gives you the option of installing a preconfigured starter database, networking services, and Oracle8i utilities. This type of installation is recommended for users who want a minimum database package.
Custom	Allows you to selectively install any product from the CD-ROM.

As a part of installation, Net8 products are also installed. This section covers the following Net8 installation-related topics:

- [Net8 Server Installation](#)
- [Net8 Client Installation](#)
- [Net8 Products Overview](#)

See your operation system's installation guide for further information about the Oracle Universal Installer, the installation process, and the install types.

Net8 Server Installation

The following products configure default information in the configuration files during Net8 Server installation:

- [Net8 Configuration Assistant](#)
- [Oracle Database Configuration Assistant](#)

Net8 Configuration Assistant

During server installation, the Net8 Configuration Assistant creates configuration files in the following manner:

Installation Type	Installation Type Results
Typical	<p>The Net8 Configuration Assistant runs without user input and configures a default:</p> <ul style="list-style-type: none">■ LISTENER.ORA file for a listener named LISTENER listening on the following endpoints:<ul style="list-style-type: none">-IPC for connections to external procedures-TCP/IP for a Net8 client connection to the Oracle database-TCP/IP for a IIOP client connection to the Java optionStatic information about external procedures is also configured in the LISTENER.ORA file.■ TNSNAMES.ORA file with a net service name configured for external procedures■ SQLNET.ORA with parameters set for the computer's domain and the naming method order
Minimal	<p>The Net8 Configuration Assistant runs without user input and configures a default:</p> <ul style="list-style-type: none">■ LISTENER.ORA file for a listener named LISTENER listening on the following endpoints:<ul style="list-style-type: none">-IPC for connections to external procedures-TCP/IP for a Net8 client connection to the Oracle database-TCP/IP for a IIOP client connection to the Java optionStatic information about external procedures is also configured in the LISTENER.ORA file.■ TNSNAMES.ORA file with a net service name configured for external procedures■ SQLNET.ORA with parameters set for the computer's domain and the naming method order

Installation Type	Installation Type Results
Custom	<p>The Net8 Configuration Assistant prompts you to configure the following:</p> <ul style="list-style-type: none"> ■ a listener, including it's name and protocols to use for listener endpoints ■ naming methods to resolve net service names to network addresses <p>When configuring the listener to listen on TCP/IP for typical Net8 clients, you should choose the default port or 1521. If you do not, you must configure the LOCAL_LISTENER parameter in the INITSID.ORA file and resolve the listener name through a naming method. For further information about configuring non-default port numbers in LISTENER.ORA for Net8 clients, see "Configuring a Listener that Uses a Non-Default Port" on page 6-67.</p> <p>When configuring the listener to listen on TCP/IP for IIOP clients, you should choose the default port of 2481. Likewise, you should the default port of 2482 for TCP/IP with SSL for IIOP clients. If you do not, you must configure the LOCAL_LISTENER parameter or the MTS_DISPATCHERS parameter's LISTENER attribute in the INITSID.ORA file, and resolve the listener name through a naming method. For further information about configuring non-default port numbers in LISTENER.ORA for IIOP clients, follow the steps in "Configuring a Non-Default Listener" on page 7-59.</p>

Oracle Database Configuration Assistant

During Typical and Minimal server installation, the Oracle Database Configuration Assistant adds information about the database to the LISTENER.ORA file. This information is required for the Oracle Intelligent Agent when using Oracle Enterprise Manager.

See "[Configuring the Listener](#)" on page 6-62 for further information about the default listener configuration.

Net8 Client Installation

For each client installation type, the Net8 Configuration Assistant runs without user input and configures configuration files in the following manner:

Installation Type	Installation Type Results
Typical	<p>The Net8 Configuration Assistant runs without user input and configures a default:</p> <ul style="list-style-type: none">■ TNSNAMES.ORA file with a net service name configured for external procedures■ SQLNET.ORA with parameters set for the computer's domain and the naming method order
Minimal	<p>The Net8 Configuration Assistant runs without user input and configures a default:</p> <ul style="list-style-type: none">■ TNSNAMES.ORA file with a net service name configured for external procedures■ SQLNET.ORA with parameters set for the computer's domain and the naming method order
Custom	<p>The Net8 Configuration Assistant prompts you to specify naming methods to resolve net service names to network addresses.</p> <p>See Chapter 6, "Configuring Naming Methods and the Listener" for further information about configuring net service names.</p>

Net8 Products Overview

The following table lists the Net8 products, the machine it should be installed on, and the installation type that installs it:

Product	Description	Located on...	Installable with the following installation type...
Net8 Client	Provides products that allow client connections to databases across a network. A client-side application sends a request to Net8 to be transported across the network to the server.	server and client	All
Net8 Server	Provides products that allow the network listener, through protocol, to accept connections from client applications on the network.	server	All
Support for the following protocols:			
■ TCP/IP	Enables client/server conversation over a network using TCP/IP. This combination of Oracle products enables an Oracle application on a client to communicate with remote Oracle databases through TCP/IP (if the Oracle database is running on a host system that supports network communication using TCP/IP).	server and client	All
■ TCP/IP with SSL	Enables client/server conversation over a network using TCP/IP and the Secure Sockets Layer (SSL). This combination of Oracle products enables an Oracle application on a client to communicate with remote Oracle databases through TCP/IP and SSL (if the Oracle database is running on a host system that supports network communication using TCP/IP and SSL).	server and client	All
■ SPX	Enables client/server conversation over a network using SPX/IPX. This combination of Oracle products enables an Oracle application on a client to communicate with remote Oracle databases through SPX/IPX (if the Oracle database is running on a host system that supports network communication using SPX/IPX). This protocol is predominantly used in Novell Netware environments.	server and client	Custom

Product	Description	Located on...	Installable with the following installation type...
<ul style="list-style-type: none"> ■ Named Pipes 	<p>Enables client/server conversation over a network using Named Pipes. This combination of Oracle products enables an Oracle application on a client to communicate with remote Oracle databases through Named Pipes (if the Oracle database is running on a host system that supports network communication using Named Pipes).</p> <p>The Named Pipes is a high-level interface providing interprocess communications between clients and servers (distributed applications). One process (the server side of the application) creates the pipe, and the other process (the client side) opens it by name. What one side writes, the other can read, and vice versa. Named Pipes is specifically designed for PC LAN environments.</p>	server and client	Custom
<ul style="list-style-type: none"> ■ Logical Unit Type 6.2 (LU6.2) 	<p>The protocol is part of the IBM Advanced Program-to-Program Communication (APPC) architecture.</p> <p>APPC is the IBM peer-to-peer (program-to-program) protocol for a System Network Architecture (SNA) network. SNA is an IBM reference model similar to the Open Systems Interconnect (OSI) model of the International Standards Organization (ISO).</p> <p>APPC architecture lets the client and host communicate over an SNA network without forcing the client to emulate a terminal (as in terminal-to-host protocols). APPC architecture allows peer-to-peer communication; the client can initiate communication with the server.</p> <p>An SNA network with the LU6.2 and Physical Unit Type 2.1 (PU2.1) protocols provides APPC. The LU6.2 protocol defines a session between two application programs; LU6.2 is a product-independent LU-type.</p> <p>LU6.2 enables an Oracle application on a PC to communicate with an Oracle database. This communication occurs over an SNA network with the Oracle database on a host system that supports APPC.</p>	server and client	Custom

Product	Description	Located on...	Installable with the following installation type...
<ul style="list-style-type: none"> ■ Bequeath 	<p>Enables clients to retrieve information from the database without using the network listener. The Bequeath protocol internally spawns a server thread for each client application. In a sense, it does the same operation that a remote network listener does for your connection, yet locally.</p> <p>Bequeath:</p> <ul style="list-style-type: none"> ■ Does not use a network listener (therefore, no network listener configuration is required) ■ Is used for local connections where an Oracle client application (such as SQL*Plus) communicates with an Oracle server running on the <i>same</i> machine ■ Only works in Dedicated Server mode. It cannot be used in a Multi-Threaded Server (MTS) mode. ■ Is supported on Windows NT for 32-bit applications. It also works on Windows 95 for connections to a Personal Oracle8i database for 32-bit applications. ■ Is installed by default with Net8 Client 	server and client	not applicable (installed with Net8 Client and Net8 Server)

Product	Description	Located on...	Installable with the following installation type...
Configuration Tools	<p>Net8 has a number of new features that simplify configuration and administration of the Oracle network:</p>		
<ul style="list-style-type: none"> ■ Net8 Configuration Assistant 	<p>Net8 Configuration Assistant configures necessary client and server Net8 components after an install, as described in:</p> <ul style="list-style-type: none"> ■ "Net8 Server Installation" on page 4-22 ■ "Net8 Client Installation" on page 4-24 <p>It may be also run in stand-alone mode to configure naming methods and the listener. To start the Net8 Configuration Assistant:</p> <ul style="list-style-type: none"> ■ On UNIX, run <code>netca</code> at <code>\$ORACLE_HOME/bin</code>. ■ On Windows NT, choose Start > Programs > Oracle - <code>HOME_NAME</code> > Network Administration > Net8 Configuration Assistant. <p>Additional Information: See:</p> <ul style="list-style-type: none"> ■ "Configuring the Network with the Local Naming Method" on page 6-12 ■ Chapter 7, "Enabling Optional Net8 Features" 	server and client	All
<ul style="list-style-type: none"> ■ Net8 Easy Config 	<p>Net8 Easy Config enables you to set up net service names in the TNSNAMES.ORA file. The functionality of this product has been added to the Net8 Configuration Assistant. Oracle Corporation recommends using the Net8 Configuration Assistant.</p>	server and client	not applicable (installed with the Oracle Net8 Assistant)

Product	Description	Located on...	Installable with the following installation type...
<ul style="list-style-type: none"> ■ Net8 Assistant 	<p>The Net8 Assistant allows you to administer the following networking components:</p> <ul style="list-style-type: none"> ■ Profile — Create or modify the client SQLNET.ORA configuration file on the local machine to control such client functions as authentication, domains, logging, tracing, and security. ■ Net Service Names— Create or modify net service names in the TNSNAMES.ORA local naming file. ■ Listeners — Create or modify the LISTENER.ORA file to control the listener. ■ Oracle Names Servers — Create or modify a centralized Oracle Names server to store networking objects, such as database addresses and net service names. An Oracle Names server may also be administered locally or remotely for such functions as starting and stopping with the Net8 Assistant. <p>To start the Net8 Assistant:</p> <ul style="list-style-type: none"> ■ On UNIX, run <code>netasst</code> at <code>\$ORACLE_HOME/bin</code>. ■ On Windows NT, choose Start > Programs > Oracle - <i>HOME_NAME</i> > Network Administration > Net8 Assistant. <p>Additional Information: See Chapter 6, "Configuring Naming Methods and the Listener" and Chapter 7, "Enabling Optional Net8 Features".</p>	server and client	All

Product	Description	Located on...	Installable with the following installation type...
Oracle Names	<p>Oracle Names is a distributed naming service developed for Oracle environments to help simplify the setup and administration of global, client/server computing networks. Oracle Names does this by establishing and maintaining an integrated system of Names servers. Oracle Names servers work like a directory service storing addresses for all the database services on a network and making them available to clients wishing to make a connection.</p> <p>Additional Information: See "Configuring the Network with the Oracle Names Method" on page 6-21.</p>	Oracle Names server machine	Custom
Oracle Connection Manager	<p>The Oracle Connection Manager provides three capabilities for Oracle8i:</p> <ul style="list-style-type: none"> ■ Connection Concentration — Allows you to multiplex multiple logical client sessions through a single transport connection to an Oracle database destination. For more information, see "Enabling Connection Concentration" on page 7-31. ■ Net8 Access Control (also known as <i>firewall support</i>) — Provides a proxy for denying or allowing access to a database server. For more information, see "Enabling Net8 Access Control" on page 7-38. ■ Multi-Protocol Support — Allows client and database servers operating on different protocol stacks to communicate with each other. For more information, see "Enabling Multi-Protocol Support" on page 7-35. 	Oracle Connection Manager machine	Custom

Product	Description	Located on...	Installable with the following installation type...
External Naming Services	Net8 offers the following external naming methods:		
<ul style="list-style-type: none"> ■ NetWare Directory Service (NDS) External Naming 	<p>NDS External Naming allows you to use external NDS naming conventions to connect to an Oracle database on a Novell NDS-enabled network.</p> <p>Additional Information: See "Configuring the Network with the External Naming Method" on page 6-51.</p>	server and client	Custom
<ul style="list-style-type: none"> ■ Network Information Service (NIS) External Naming 	<p>Organizations and corporations already using Network Information Service (NIS) as part of their systems infrastructure have the option to store Oracle service aliases and addresses in NIS, using NIS External Naming</p> <p>Additional Information: See "Configuring the Network with the External Naming Method" on page 6-51.</p>	server and client	Custom
Oracle Advanced Security	Oracle Advanced Security consists of the following components:	server and client	Custom
<ul style="list-style-type: none"> ■ Network Security 	<p>This Oracle network data encryption and checksumming service ensures secure transmission of data over networks. Network Security uses encryption and authentication engines from RSA Data Security, Incorporated.</p> <p>The following algorithms are supported:</p> <p>Encryption</p> <ul style="list-style-type: none"> ■ RC4_40 (US, Export & Upgrade) ■ RC4_56 (US & Export) ■ RC4_128 (US only) ■ DES_56 (US & Upgrade) ■ DES_40 (US, Export & Upgrade) <p>Checksumming</p> <ul style="list-style-type: none"> ■ MD5 (US, Export & Upgrade) 	server and client	Custom

Product	Description	Located on...	Installable with the following installation type...
<ul style="list-style-type: none"> ■ Single Sign-On 	<p>Allows users to access multiple accounts and applications with a single password. This feature eliminates the need for multiple passwords for users and simplifies management of user accounts and passwords for system administrators.</p> <p>Centralized, secure authentication services allow you to have high confidence in the identity of users, clients, and servers in distributed environments. Network authentication services can also provide the benefit of single sign-on for users.</p> <p>The following authentication methods are supported:</p> <ul style="list-style-type: none"> ■ CyberSAFE ■ Identix (Biometric) ■ Kerberos ■ RADIUS ■ SecurID ■ SSL 	<p>server and client</p>	<p>Custom</p>
<ul style="list-style-type: none"> ■ DCE Integration 	<p>Distributed Computing Environment (DCE) Integration enables users to transparently use Oracle tools and applications to access Oracle8i databases in a DCE environment. The Oracle DCE Integration product consists of two major components:</p> <ul style="list-style-type: none"> ■ DCE Communications/Security method ■ DCE CDS (Cell Directory Service) Naming method 	<p>server and client</p>	<p>Custom</p>

Considerations for Upgrades and Migration

This chapter describes coexistence of SQL*Net and Net8, migration and upgrades. Specific topics discussed are:

- [Net8 and SQL*Net Differences](#)
- [Client and Database Compatibility Issues](#)
- [Using the Net8 Assistant to Handle Compatibility Issues](#)
- [Migrating and Upgrading Considerations](#)
- [Upgrading and Migrating to Net8 Release 8.1](#)
- [Additional Considerations for Oracle Connection Manager and Oracle Names](#)

Net8 and SQL*Net Differences

When dealing with SQL*Net to Net8 migration, keep the following features in mind:

This Feature...	SQL*Net	Net8
Multiple Protocol Support	Multi-Protocol Interchange allows applications in TNS networks to communicate across different protocols.	The Oracle Connection Manager has a multi-protocol support feature which allows client and database servers operating on different protocol stacks to communicate with each other. Additional Information: See "Enabling Multi-Protocol Support" on page 7-35.
Configuration Tools	SQL*Net provides: <ul style="list-style-type: none"> ■ SQL*Net Easy Configuration to create service names ■ Oracle Network Manager for complex configurations 	Net8 provides: <ul style="list-style-type: none"> ■ Net8 Configuration Assistant for basic configuration of the listener, net service names, and naming methods. ■ Net8 Easy Config to create, modify, delete, or test net service names ■ Net8 Assistant for complex configurations and network administration
External Procedures	n/a	Support for external procedure, a function or procedure written in a third-generation language (3GL) that can be called from PL/SQL code.

Though it is not required, migrating your network to Net8 is recommended for the following reasons:

- Minimal configuration.

Net8 simplifies the process of setting up your network components. With Net8, you can start a client, network listener, Oracle Names server, and Oracle Connection Manager with default settings. This minimizes the need to create and maintain configuration files.

As a part of the installation process, the Net8 Configuration Assistant configures the basic parameters in the Net8 configuration files needed for simple environments.

With Net8, the Net8 Assistant replaces most of the functionality previously provided with Oracle Network Manager. Use the Net8 Assistant to create or modify your existing TNSNAMES.ORA local naming files, SQLNET.ORA profiles, LISTENER.ORA listener files, and NAMES.ORA Oracle Names configuration files.

- Extended network functionality. Net8 has replaced many of the features previously available with SQL*Net version 2 with equivalent or enhanced functionality.

The table below lists the networking features supported in each Oracle release.

Database Release	7.1.4	7.1.5	7.1.6	7.2.2	7.2.3	7.3.2	7.3.3	7.3.4	8.x
SQL*Net	2.1.4	2.1.5	2.1.6	2.2.2	2.2.3	2.3.2	2.3.3	2.3.4	n/a
Oracle Names server	1.0	1.0	1.0	1.1	2.0	2.0.2	2.0.3	2.0.4	8.x
Oracle Advanced Networking Option	n/a	n/a	n/a	n/a	n/a	2.3.2	2.3.3	2.3.4	8.0.x
Oracle Advanced Security	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	8.1.x
Secure Network Services ¹	1.0.1	1.0.2	1.0.3	1.1	2.0	n/a	n/a	n/a	n/a
Net8	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	8.x

¹ The functionality of Secure Network Services and SQL*Net/DCE are now included in the Oracle Advanced Networking Option and Oracle Advanced Security option.

Obsolete SQL*Net Parameters

The following SQL*Net parameter are no longer functional in Net8:

Parameter	File	Description
COMMUNITY	TNSNAMES.ORA	The COMMUNITY parameter used to a required part of all network service addresses. Thus, it appears anywhere you might find an address (for example, local naming and listener configuration files).
NAMES.DEFAULT_ZONE	SQLNET.ORA	The NAMES.DEFAULT_ZONE used to be included in profiles as slight variants of the NAMES.DEFAULTDOMAIN and NAMES.PREFERRED_SERVERS parameters.

You may choose to delete these parameters or leave them. Though they are no longer required, they will not interfere with any network operations.

Client and Database Compatibility Issues

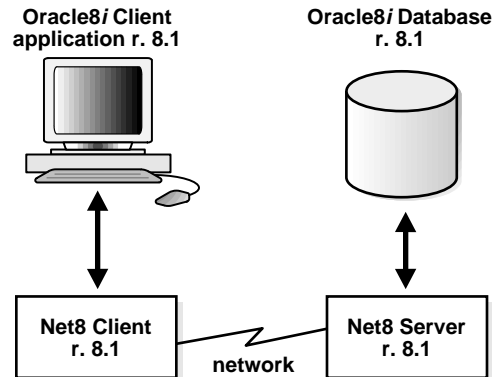
Consider the following client-to-database connection issues before you decide if upgrading or migrating is appropriate for you environment:

- [Oracle8i Client Release 8.1 Connection to Oracle8i Database Release 8.1](#)
- [Oracle8i Release 8.0/Oracle7 Client Connection to Oracle8i Database Release 8.1](#)
- [Oracle8i Client Release 8.1 Connection to Oracle8i Database Release 8.0/Oracle7 Database](#)
- [Oracle Names](#)

Oracle8i Client Release 8.1 Connection to Oracle8i Database Release 8.1

As [Figure 5-1](#) depicts, an Oracle8i Client release 8.1 requires Net8 Client release 8.1, and an Oracle8i release 8.1 database requires Net8 Server release 8.1:

Figure 5-1 Oracle8i Client to Oracle8i Database

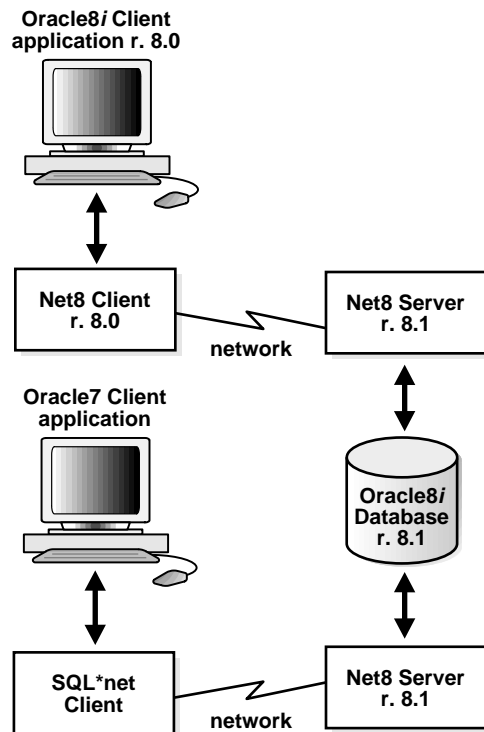


The Oracle8i Client must be configured with a service name, as described in ["Configuring the Network with the Local Naming Method"](#) on page 6-12.

Oracle8i Release 8.0/Oracle7 Client Connection to Oracle8i Database Release 8.1

As [Figure 5-2](#) depicts, an Oracle8i Client release 8.0 requires a compatible release of Net8 Client, an Oracle7 Client requires SQL*Net Client, and an Oracle8i database requires Net8 Server release 8.1.

Figure 5-2 Oracle8i Client release 8.0/Oracle7 Client to Oracle8i Database



While is *not* necessary to replace the Oracle System Identifier (SID) of the database with its service name, Oracle recommends doing so to take advantage of new functionality. For example:

```
net_service_name=
(description=
  (address=...)
  (address=...)
  )
(connect_data=
  (service_name=sales.com)===
  )
```

This effect may be accomplished by replacing `SID=SID` with `SERVICE_NAME=SERVICE_NAME` with the Net8 Assistant's compatibility mode. See ["Using the Net8 Assistant to Handle Compatibility Issues"](#) on page 5-11.

Consider the following questions for an environment with Oracle7 clients connecting to an Oracle8i release 8.1 database:

- Will my third party Oracle7 applications be able to take advantage of Net8 features?

No. You must rebuild or upgrade applications to work with Net8 libraries.
- *Do my Oracle7 clients require Net8 Client to connect to a remote Oracle8i database?*

No. If an Oracle7 client needs to connect to a *remote* Oracle8i database, only SQL*Net Client release 2.x has to be configured on the Oracle7 client. Net8 is backward compatible with SQL*Net release 2.x. The only limitation is that the new network features available with Net8 are unavailable with this connection type.
- *Do my Oracle7 clients require Net8 Client to connect to a local Oracle8i database?*

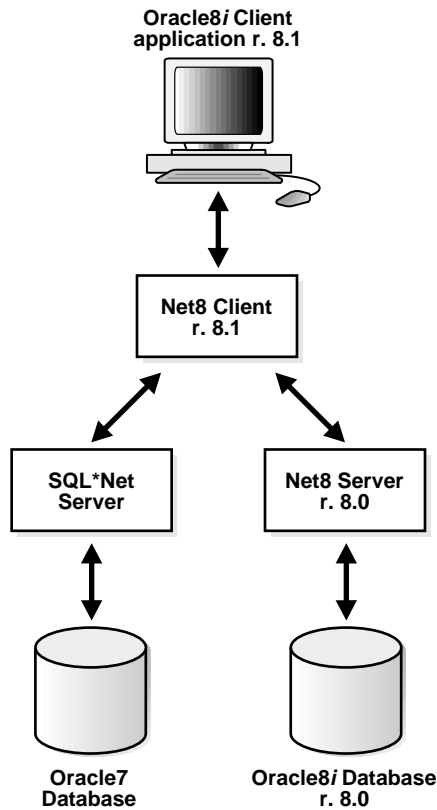
Yes. If the Oracle7 client needs to connect to a *local* Oracle8i database, you have SQL*Net Client release 2.x, Net8 Client, and Net8 Server in the same system. Note that Net8 Client and Net8 Server are already installed during the installation of Oracle8i Enterprise Edition or Oracle8i.

Oracle8i Client Release 8.1 Connection to Oracle8i Database Release 8.0/Oracle7 Database

As depicted in [Figure 5-3](#):

- An Oracle8i Client release 8.1 requires Net8 Client release 8.1.
- An Oracle7 Server requires SQL*Net Server.
- An Oracle8i database release 8.0 requires Net8 Server release 8.0.

Figure 5-3 Oracle8i Client to Oracle8i Release 8.0/Oracle7 Database



The release 8.1 clients must be configured with the SID of the database. For example, the TNSNAMES.ORA would have the following effect:

```
net_service_name=  
(description=  
  (address=...)   
  (address=...)   
  )   
(connect_data=  
  (sid=sales)   
  )
```

This effect may be accomplished with the Net8 Assistant's compatibility mode. See ["Using Release 8.0/7.x Features and Connecting To a Release 8.0/7.x Service"](#) on page 5-14.

Additionally, the LISTENER.ORA on the database server must still be configured with the description of the SID, as described in ["Configuring a Listener with Service Information"](#) on page 6-71.

Consider the following questions for an environment with Oracle8i clients release 8.1 connecting to an Oracle7 database.

- *Do my Oracle8i clients require SQL*Net Client release 2.x to connect to a remote Oracle7 database?*

No. If an Oracle8i client needs to connect to a *remote* Oracle7 database, only Net8 Client needs to be configured on the Oracle8i client. SQL*Net release 2.x is upwards compatible with Net8. The only limitation is that the new network features available with Net8 are unavailable with this connection type.

- *Do my Oracle8i clients require SQL*Net Client release 2.x to connect to a local Oracle7 database?*

Yes. If the Oracle8i client needs to connect to a *local* Oracle7 database, you need both SQL*Net Server release 2.x and Net8 Client on the same system.

Oracle Names

If you are migrate or upgrade all or part of your network to Net8, you should migrate or upgrade all the Oracle Names Servers in the region to version 8.

- *Can my Oracle7 clients use Oracle Names version 8 to resolve service names?*

Yes.

- *Can my Oracle7 client then use the connect string returned form Oracle Names version 8 to connect to an Oracle7 or Oracle8i database?*

Yes, if the connect string was specified correctly when it was entered into Oracle Names. In other words, if the connect string worked from a TNSNAMES.ORA file, it will work from Oracle Names.

Using the Net8 Assistant to Handle Compatibility Issues

Because some parameters are enabled only for release 8.1, the Net8 Assistant offers two options that permit you set the proper parameters in the TNSNAMES.ORA file for Net8 release 8.1 clients or Net8 release 8.0/SQL*Net release 2.x clients:

Net8 Assistant option	Description
<i>Use Options Compatible with Net8 8.0 Clients</i>	<p>Allows you to enable or disable release 8.1 options for multiple addresses.</p> <p>If turned ON, allows you to only select the Oracle Connection Manager parameter, SOURCE_ROUTE, for pre-release 8.1 client connections.</p> <p>If turned OFF, allows you to enable or disable SOURCE_ROUTE, LOAD_BALANCE and FAILOVER parameters for release 8.1 client connections.</p> <p>Additional Information: See "Configuring Multiple Address Options" on page 7-11.</p>
<i>Use Oracle8i Release 8.0 Compatible Identification</i>	<p>Allows you to enable or disable release 8.1 advanced options for the CONNECT_DATA section of the TNSNAMES.ORA.</p> <p>If turned ON, allows you to enter the SID of the pre-release 8.1 database.</p> <p>If turned OFF, allows you to enter the release 8.1 service name (SERVICE_NAME).</p> <p>Note: The <i>Advanced Service Options</i> dialog box, which is visible when the Advanced button in the Service Identification group is chosen, is also impacted by whether this option is turned ON or OFF. Some settings are only available for connections to a release 8.1 service.</p> <p>Additional Information: See "Configuring Advanced Connection Attributes for a Net Service Name" on page 7-7.</p>

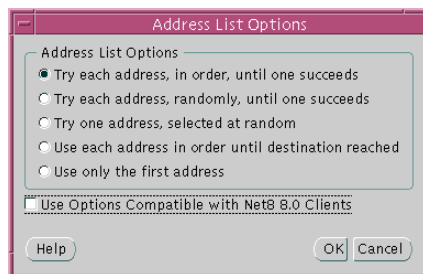
Using Release 8.1 Features and Connecting To a Release 8.1 Service

If using older configuration files prior to release 8.1, the TNSNAMES.ORA may be modified to take advantage of new features or to connect to a release 8.1 service. These changes can be made manually to the TNSNAMES.ORA file or with the Net8 Assistant.

To configure old configuration files for release 8.1 features, use either the Net8 Assistant:

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Double-click on the Net Service Names folder in the directory tree.
3. Select a net service name for which you want the compatibility mode to apply.
4. If multiple addresses are configured for the selected net service name, you may configure client load balancing, connect-time failover and Oracle Connection Manager options:
 - a. Click the Advanced button in the Address Configuration group.

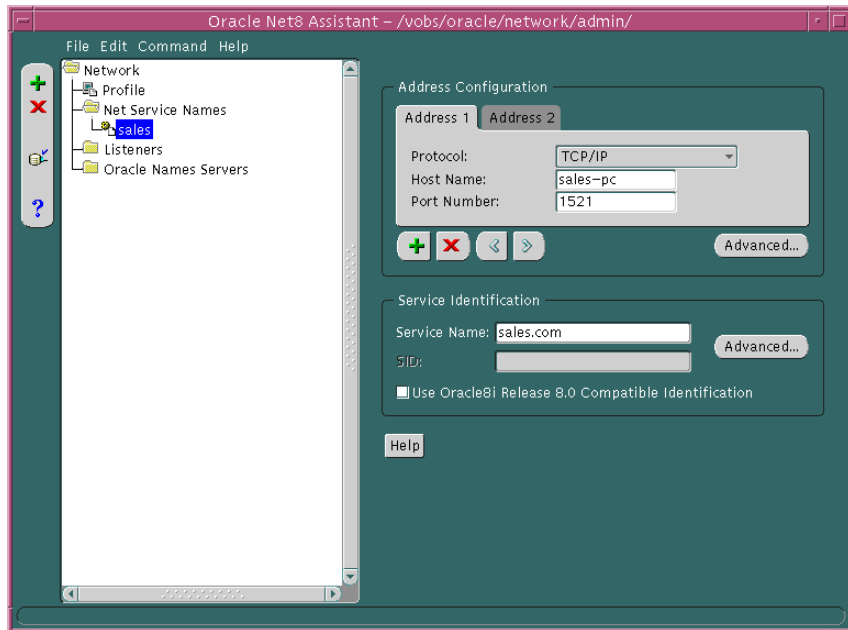
The *Address List Options* dialog box appears:



- b. Ensure *Use Options Compatible with Net8 8.0 Clients* is turned OFF.
- c. Select any of the address list options, then click OK.

See "[Configuring Multiple Address Options](#)" on page 7-11 for further information about the options.

5. Connect to a release 8.1 service:



- a. Ensure *Use Oracle8i Release 8.0 Compatible Identification* is turned OFF in the Service Identification group box.
- b. Enter the name of the service to which you want to connect in the Service Name field.

The service name is typically the global database name, a name comprised of the database name and domain name, entered during installation or database creation.

For further information about the service name value, see:

- "[Setting Service Names and Instance Names](#)" on page 6-3 and "[Step 1: Verify Service Name and Instance Name](#)" on page 5-17 to set the service name
- "[Configuring Advanced Connection Attributes for a Net Service Name](#)" on page 7-7 for more information about optional connection settings

6. Choose Save Network Configuration from the File menu.

7. Choose Exit from the File menu.

The Net8 Assistant application exits.

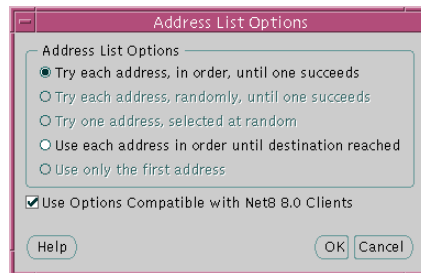
Using Release 8.0/7.x Features and Connecting To a Release 8.0/7.x Service

If configuring a client to connect to a release 8.0 or release 7.x service, the TNSNAMES.ORA must be configured with the SID of database.

To configure release 8.0 features, use either the Net8 Assistant or manually modify the TNSNAMES.ORA file:

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Double-click on the Net Service Names folder in the directory tree.
3. Select a net service name for which you want the compatibility mode to apply.
4. If multiple addresses are configured for the selected net service name, you may configure Oracle Connection Manager:
 - a. Click the Advanced in the Address Configuration group.

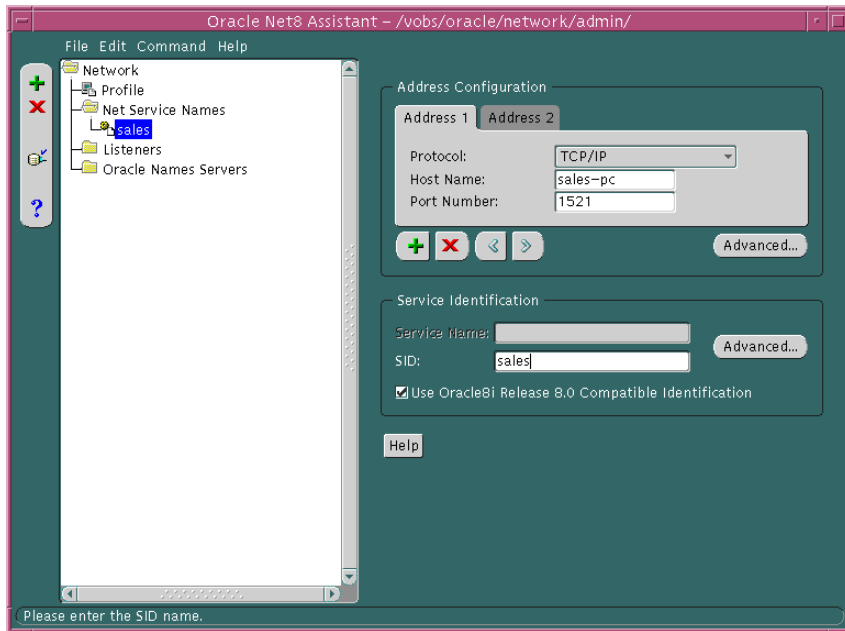
The *Address List Options* dialog box appears:



- b. Ensure *Use Options Compatible with Net8 8.0 Clients* is turned ON.
- c. Select *Use each address in order until destination reached*. The *Try each address, in order, until one succeeds* option turns the source route option OFF. The other options are connect-time failover and client load balancing options that are not available for pre-release 8.1 clients.

See "[Configuring the Client for Oracle Connection Manager Features](#)" on page 7-41 for further information about configuring Oracle Connection Manager.

5. Connect to a release 8.0 or release 7.x service:



- a. Ensure *Use Oracle8i Release 8.0 Compatible Identification* is turned ON.
 - b. Enter the name of the SID to which you want to connect in the SID field.
 6. Choose Save Network Configuration from the File menu.
 7. Choose Exit from the File menu.
- The Net8 Assistant application exits.

Migrating and Upgrading Considerations

When installing Net8, you have these installation decisions to make:

Installation Decision	Description
Install into another Oracle home	Install release 8.1.5 in a different Oracle home on the same server.
Migrate	Migrating is the process of moving from SQL*Net release 2.x to Net8. Migration involves installing Net8 and de-installing SQL*Net release 2.x.
Upgrade	Upgrading is the process of transforming <i>one release to another release of the same database version</i> . For example, transforming an Net8 release 8.0.5 to an Oracle8i database release 8.1.5 is upgrading the database system.

Note: It is not possible to install release 8.1.5 products into an existing Oracle home.

Upgrading and Migrating to Net8 Release 8.1

To upgrade from Net8 release 8.0 to Net8 release 8.1 or migrate from SQL*Net release 2.x to Net8 release 8.1, follow these tasks:

[Step 1: Verify Service Name and Instance Name](#)

[Step 2: Perform Software Upgrade or Migration on the Server](#)

[Step 3: Perform Software Upgrade or Migration on the Client](#)

[Step 4: Perform Functional Upgrade and Migration](#)

Step 1: Verify Service Name and Instance Name

If you want to take advantage of the Net8 release 8.1 feature to identify a service and its instance in the TNSNAMES.ORA file, ensure the SERVICE_NAMES and INSTANCE_NAMES parameters are set in the database initialization file (INITSID.ORA).

Parameter	Description
SERVICE_NAMES	One or more comma separated strings that represent the names of the services on the network. One of the services should be set to the global database name, a name comprised of the database name (DB_NAME) and domain name (DB_DOMAIN).
INSTANCE_NAME	A string value representing the name of instance and is used to uniquely identify a specific instance when multiple instances share common services names. INSTANCE_NAME should not be confused with the SID, which actually uniquely identifies the instances shared memory on a host. The INSTANCE_NAME should be set to the value of the SID.

Step 2: Perform Software Upgrade or Migration on the Server

Note: You cannot install release 8.1.5 into an Oracle home that was created using an earlier release of Oracle8*i* database and you cannot install releases of the Oracle8*i* database prior to release 8.1.5 into an Oracle home that was created by release 8.1.5.

To perform a software upgrade or migration on the database server:

Install the latest release of Net8 Server from the Oracle Universal Installer to receive the latest executables.

If Net8 Server is installed with the Oracle8*i* database, you are prompted to upgrade a database with the Oracle Data Migration Assistant if the Oracle Universal Installer detects a release 8.0 database on your system. If you do not want to upgrade during the installation process, you can choose to install this assistant and use it later.

The Oracle Universal Installer automatically performs these tasks:

- Net8 release 8.0 or SQL*Net listener is stopped.
- Net8 release 8.1 listener is started.

Step 3: Perform Software Upgrade or Migration on the Client

To perform a software upgrade or migration on the client

Install the latest release of Net8 Client from the Oracle Universal Installer to receive the latest executables.

Step 4: Perform Functional Upgrade and Migration

After the software is upgraded and migration, it is *not* required to upgrade the configuration files unless you want to use the new release 8.1 features. To take advantage of new features:

For this configuration file...	Perform the following...
SQLNET.ORA on clients and server	Remove AUTOMATIC_IPC, the parameter to force session to use IPC, as it will be ignored.
TNSNAMES.ORA	<p>Replace SID with the SERVICE_NAME parameter to connect to a release 8.1 service.</p> <pre>net_service_name= (description= (address=...) (address=...)) (connect_data=(service_name=sales.com))</pre> <p>The SERVICE_NAME is typically the global database name, a name comprised of the database name and domain name, entered during installation or database creation.</p> <p>Additional Information: See:</p> <ul style="list-style-type: none"> ▪ "Using the Net8 Assistant to Handle Compatibility Issues" on page 5-11 for information about configuring the service name ▪ "Setting Service Names and Instance Names" on page 6-3 for information about the service name string
	<p>Enable client load balancing and/or connect-time failover.</p> <p>Additional Information: See:</p> <ul style="list-style-type: none"> ▪ "Using the Net8 Assistant to Handle Compatibility Issues" on page 5-11 to set LOAD_BALANCE and/or FAILOVER for an ADDRESS_LIST. ▪ "Configuring Multiple Address Options" on page 7-11

For this configuration file... Perform the following...

LISTENER.ORA

Because instance information is registered with the listener in release 8.1, it is no longer necessary to include the instance information with the `SID_LIST_listener_name` section of the LISTENER.ORA file.

However, many management tools, including Oracle Enterprise Manager, still require this information for release 8.1.5. If you are using Oracle Enterprise Manager to manage database objects, the LISTENER.ORA file must be configured with information about the database in the following manner:

```
sid_list_listener_name=  
  (sid_list=  
    (sid_desc=  
      (global_dbname=global_database_name)  
      (oracle_home=oracle_home)  
      (sid_name=sid)  
    )  
  )  
)
```

Implementing connect-time failover does not allow use of static service configuration parameters in the LISTENER.ORA file. However, static configuration is required for Oracle8i release 8.0 or Oracle7 databases and Oracle Enterprise Manager. For further information about statically configuring the listener, see "[Configuring a Listener with Service Information](#)" on page 6-71.

See the *Oracle8i Parallel Server Setup and Configuration Guide* for additional configuration instructions on setting up connect-time failover for Oracle Parallel Server in an Oracle Enterprise Manager environment.

Additional Information: See "[Configuring a Listener with Service Information](#)" on page 6-71

See the next section, "[Additional Considerations for Oracle Connection Manager and Oracle Names](#)", for additional Oracle Connection Manager and Oracle Names migration issues.

Additional Considerations for Oracle Connection Manager and Oracle Names

If you are currently using Oracle Multi-Protocol Interchange or Oracle Names, review the following topics:

- [Using Oracle Connection Manager Release 8.1](#)
- [Using Oracle Connection Manager Instead of Oracle Multi-Protocol Interchange](#)
- [Using Oracle Names Version 8](#)

Using Oracle Connection Manager Release 8.1

The Oracle Connection Manager's administrative process, CMADMIN, is used by the CMCTL control utility to execute administrative commands.

In release 8.1, new configurations require a protocol address for CMADMIN. By default this address is:

```
cman_admin=  
  (address=  
    (protocol=tcp)  
    (host= host)  
    (port=1830)  
  )
```

Oracle Connection Manager release 8.1 is backwards compatible with release 8.0.

This implies that:

- Oracle Connection Manager release 8.0 can access an Oracle8i release 8.1 database. If the old CMAN.ORA file is being used, the CMADMIN process will listen on TCP/IP with port 1830. If port 1830 is in use, manually configure CMAN.ORA with another CMAN_ADMIN address.
- Oracle Connection Manager release 8.1 can access an Oracle8i release 8.0 database. Functionality, however, reverts to release 8.0.
- Oracle8i release 8.0 clients can access Oracle Connection Manager release 8.1 without any additional configuration.

Using Oracle Connection Manager Instead of Oracle Multi-Protocol Interchange

If you have migrated your clients to Net8, and you still require multi-protocol support, you will need to install Oracle Connection Manager, and route your sessions through it. Oracle Multiprotocol Interchange is no longer supported past SQL*Net release 2.3.

To ensure proper functioning of Oracle Connection Manager, verify that you have deleted the following files on those nodes:

- TNSNET.ORA
- TNSNAV.ORA
- INTCHG.ORA

Other migration considerations are specific to your network configuration.

See "[Enabling Multi-Protocol Support](#)" on page 7-35, for further information about multi-protocol support.

Using Oracle Names Version 8

Oracle Names version 8 is backward compatible with SQL*Net version 2. This implies that clients running on SQL*Net version 2 can access Oracle Names servers using Oracle Names version 8 to connect to an Oracle8i database.

If you wish to take advantage of the new features provided with Oracle Names version 8, you must migrate all of your existing Names Servers in a region to version 8 by installing Oracle Names version 8 on every existing Names Server node.

Migration issues to keep in mind include:

- [Migrating from Oracle Names Version 2 Using a Database](#)
- [Migrating from Oracle Names Version 2 Using the Dynamic Discovery Option](#)
- [Migrating from ROSFILES to Oracle Names Tables in a Database](#)
- [Checklist for Ensuring Proper Migration to Oracle Names Version 8](#)

Migrating from Oracle Names Version 2 Using a Database

To migrate and transfer data from an existing Oracle Names server database to a version 8 database, run the NAMESUPG.SQL script located in `$ORACLE_HOME/network/admin` on UNIX and `ORACLE_HOME\network\admin` on Windows NT on the node where Oracle Network Manager stored your network definition:

```
sql> CONNECT user/password
sql> @oracle_home/network/admin/namesupg.sql;
```

Migrating from Oracle Names Version 2 Using the Dynamic Discovery Option

The procedure to migrate Oracle Names version 2 with the Dynamic Discovery Option is dependent upon whether or not you want Oracle Names version 8 to store information in a region database.

- [Non-Region Database Migration](#)
- [Region Database Migration](#)

Non-Region Database Migration If you migrate to a Oracle Names version 8 from Oracle Names version 2 with the Dynamic Discovery Option, the new Oracle Names server should be able obtain registered data from the old checkpoint files. If for some reason data is not registered, you can register objects, following the procedures in "[Step 5: Configure the Listener](#)" on page 6-40 to configure the listener properly with the USE_PLUG_AND_PLAY parameter and "[Step 6: Register Data with the Oracle Names Server](#)" on page 6-41 to register objects.

Region Database Migration If you were previously running Oracle Names version 2 using the Dynamic Discovery Option, and you want to configure a database as a repository for your Oracle Names information, you will need to:

1. Write the information stored in the Oracle Names version 2 local administrative region to a TNSNAMES.ORA file. From the command line, enter:

```
namesctl
namesctl> dump_tnsnames
```

2. Run the NAMESINI.SQL script located in \$ORACLE_HOME/network/admin on UNIX and *ORACLE_HOME*\network\admin on Windows NT on the Oracle Names Server node where you want the database to reside:

```
sql> CONNECT user/password
sql> @oracle_home/network/admin/namesini.sql;
```

3. Load the TNSNAMES.ORA file into a version 8 Oracle Names server using either the Net8 Assistant or the NAMESCTL control utility:

Use the Net8 Assistant...	Use the control utility NAMESCTL...
<ol style="list-style-type: none"> 1. Start Net8 Assistant. -On UNIX, run <code>netasst</code> at <code>\$ORACLE_HOME/bin</code>. -On Windows NT, choose Start > Programs > Oracle - <i>HOME_NAME</i> > Network Administration > Net8 Assistant. 2. Double-click on the Oracle Names server folder. 3. Select the Oracle Names server. 4. Select Manage Data from the drop-down list box. 5. Select Load. 6. Enter the path and file name of the Oracle Network Manager-generated TNSNAMES.ORA file in the File field. 7. Select Execute. 8. Select Save Network Configuration from the File menu. 9. Select Exit from the File menu to exit the Net8 Assistant application. 	<p>From the command line, enter:</p> <pre>namesctl namesctl> load_tnsnames file_name</pre>

4. Use the Net8 Assistant to configure a NAMES.ADMIN_REGION parameter in every Oracle Names server configuration file (NAMES.ORA). See ["Step 3: Create an Oracle Names Server"](#) on page 6-25.
5. Reload Oracle Names server information into at least one Oracle Names server by issuing a NAMESCTL REORDER_NS command on one of the Oracle Names server node. See ["Step 4: Configure Clients and Database Servers To Use Oracle Names Servers"](#) on page 6-37.

Migrating from ROSFILES to Oracle Names Tables in a Database

Oracle Names version 8 does not support older configurations that use Resource Object Store (ROS) files (ROSFILES). ROSFILES must be migrated directly into Oracle Names database tables or first into TNSNAMES.ORA files and then into Oracle Names.

- [ROSFILES to Database Tables](#)
- [ROSFILES to TNSNAMES.ORA File](#)

ROSFILES to Database Tables To migrate ROSFILES to database tables:

1. Create a database user account for Oracle Network Manager:

```
sql> CONNECT system/password
sql> CREATE USER user
      identified by password
      default tablespace users
      temporary tablespace temp;
```

2. Grant SYSDBA privileges to the user:

```
sql> GRANT sysdba to username identified by password;
```

3. To build the necessary tables, the scripts described below must be run against the server. Typically, these scripts are run on the Oracle Network Manager node.

```
sql> CONNECT user/password
sql> @oracle_home\dfs\rosbild.sql ;
sql> @oracle_home\dfs\nmcbild.sql ;
sql> @oracle_home\dfs\rosgrnt.sql ;
sql> @oracle_home\dfs\nmcgrnt.sql ;
```

Script	Description
ROSBILD.SQL	Builds tables for use by the ROS.
NMCBILD.SQL	Builds tables for use by the Oracle Network Manager Objects (NMO) components.
ROSGRNT.SQL	Grants access to the users who will access the common tables. You will be prompted for the user name. Use the same user name that was used when you set up the Oracle Network Manager account.
NMCGRNT.SQL	Grants access to the users who will access the Oracle Network Manager tables.

4. From the Oracle Network Manager, save the ROSFILES to a database:
- Choose Save As from the File menu.
 - Select Database in the *Save Network Definition* dialog box, then click OK.
 - Enter the database user name/password created in step 1 and a net service name for the database in the *Connect* dialog box. Click OK.
 - Select or enter the name of the network you wish to save in the *Save Network Definition* dialog box.
 - Choose Exit from the File menu to exit the Oracle Network Manager.
5. On the server, upgrade the database file by running the NAMESUPG.SQL script located in \$ORACLE_HOME/network/admin on UNIX and ORACLE_HOME\network\admin on Windows NT:

```
sql> CONNECT user/password
sql> @oracle_home/network/admin/namesupg.sql ;
```

ROFILES to TNSNAMES.ORA File To migrate ROFILES to a TNSNAMES.ORA file, and then import the TNSNAMES.ORA file into Oracle Names:

1. Create a TNSNAMES.ORA file:
 - a. From the Oracle Network Manager, choose Preferences from the Special menu.
 - b. Ensure Oracle Names is *not* selected in the *Preferences* dialog box.
 - c. Choose Generate from the File menu to update the network definition and create a TNSNAMES.ORA file.
 - d. Choose Exit from the File menu to exit the Oracle Network Manager.
2. Load the TNSNAMES.ORA into the Oracle Names server using either the Net8 Assistant or the NAMESCTL control utility:

Use the Net8 Assistant...	Use the control utility NAMESCTL...
<ol style="list-style-type: none"> 1. Start Net8 Assistant. <ul style="list-style-type: none"> -On UNIX, run <code>netasst</code> at <code>\$ORACLE_HOME/bin</code>. -On Windows NT, choose Start > Programs > Oracle - <code>HOME_NAME</code> > Network Administration > Net8 Assistant. 2. Double-click on the Oracle Names server folder. 3. Select the Oracle Names server. 4. Select Manage Data from the drop-down list box. 5. Select Load. 6. Enter the path and file name of the Oracle Network Manager-generated TNSNAMES.ORA file in the File field. 7. Select Execute. 8. Select Save Network Configuration from the File menu. 9. Select Exit from the File menu to exit the Net8 Assistant application. 	<p>From the command line, enter:</p> <pre>namesctl namesctl> load_tnsnames file_name</pre>

See also the following documentation:

- *Oracle Network Manager Administrator's Guide*, release 3.1
- *Oracle Names Administrator's Guide*, version 2

Checklist for Ensuring Proper Migration to Oracle Names Version 8

The following checklist is provided to ensure proper migration to Oracle Names version 8.

- Migrate all Oracle Names servers in each region to the same version 8.
- If you were previously running Oracle Names version 2, and you want to update your database as a repository for your Oracle Names information, you will need to run the NAMESUPG.SQL script located in `$ORACLE_HOME/network/admin` on UNIX and `ORACLE_HOME\network\admin\names` on Windows NT on the node where the network definition is stored.
- If you were previously running Oracle Names version 2 using the Dynamic Discovery Option, and you want to configure a database as a repository for your Oracle Names information, you will need to:
 1. Run the NAMESINI.SQL script located in `$ORACLE_HOME/network/admin` on UNIX and `ORACLE_HOME\network\admin\names` on Windows NT on the node where you wish to install the database.

```
sql> CONNECT user/password
sql> @oracle_home/network/admin/names/namesini.sql;
```
 2. Use the Net8 Assistant to configure a NAMES.ADMIN_REGION parameter in every Names Server configuration file. For more information about the [NAMES.ADMIN_REGION](#) parameter, see [Appendix C, "Configuration Parameters"](#).
- Set up at least two Oracle Names servers in each region to provide for fault tolerance.

- ❑ Clients contact Oracle Names server through a preferred Oracle Names server list, well-known Oracle Names server, or generated list obtained through discovery, as described in "[Differences Between Versions of Oracle Names](#)" on page 2-42 and "[Understanding Discovery](#)" on page 2-45.

If you would like to use the version 8 generated list of Oracle Names servers, .SDNS.ORA on UNIX and SDNS.ORA on Windows platforms, follow "[Step 4: Configure Clients and Database Servers To Use Oracle Names Servers](#)" on page 6-37

After the file is created, delete the NAMES.PREFERRED_SERVERS parameter in the SQLNET.ORA file. Any Oracle Names servers specified in the NAMES.PREFERRED_SERVERS parameter will override the results of the discovery process.

Configuring Naming Methods and the Listener

This chapter describes how to configure your network.

Specific topics discussed are:

- [Configuration Overview](#)
- [Setting Service Names and Instance Names](#)
- [Configuration Models](#)
- [Configuring Naming Methods and the Listener](#)
- [Configuring the Network with the Local Naming Method](#)
- [Configuring the Network with the Oracle Names Method](#)
- [Configuring the Network with the External Naming Method](#)
- [Configuring the Listener](#)
- [Configuring Protocol Addresses](#)

Configuration Overview

Before a service, such as a database, can receive connections from clients, clients must be configured to use net service names. *Net service names* are network descriptions of service names. Each network description specifies one or more network routes to the service, including any number of Net8 components, such as listeners. Net service names also specify whether network connections should be load balanced among multiple listeners and/or fail over to an alternate listener if a network connection cannot be established.

Specifically, a net service name identifies the:

- network route to the service, including the location of the listener through a protocol address
- service name(s) used to identify the release 8.1 services, or Oracle System Identifier (SID) used to identify a pre-release 8.1 databases
- optional instance name of the release 8.1 service to which to connect

The listener address is used by the client to connect to the listener. Once connected, the client passes the address and service information to the listener.

When an instance starts, it registers the following registration information with the listener:

- service registration provides the listener with instance information, such as service names and instance names
- protocol address information with which the listener may connect to the instance

When listener receives a client's request for connect, it verifies the client information with the information it has received from the instance, as well as information it has stored in the LISTENER.ORA file. If the information matches, a connection is granted.

Client configuration is accomplished by creating a list of the net service names that map to services and addresses of listener destinations stored in a TNSNAMES.ORA client configuration file, an Oracle Names server, or another naming method. When an end user connects to a service, the service is identified by the net service name. For example:

```
sql> CONNECT username/password@net_service_name
```

Setting Service Names and Instance Names

A *service name* is a logical representation of a service, which may include a database. A service may comprise one or more instances. When creating a net service name, you must identify a release 8.1 service with a service name. Optionally, you may identify a release 8.1 instance with an instance name.

Shown below is a sample of TNSNAMES.ORA file, which shows how a net service provides a network description of a service:

```
net_service_name=
(description=
  (address= (protocol_address_information))
  (connect_data=
    (service_name=service_name)
    (instance_name=instance_name)))
```

In the example above, notice how the service name is identified by the SERVICE_NAME parameter and the instance name is identified by the INSTANCE_NAME parameter.

The service name is a string that is the *global database name*, a name comprised of the database name and domain name, entered during installation or database creation. If you are not sure what the global database name is, you can obtain it from the combined values of the SERVICE_NAMES parameter in the database initialization file, INITSID.ORA. For example, a service name may be SALES.COM, where SALES is the database name and COM is the domain.

The *instance name* includes the name of the instance. It is typically the SID entered during installation. If you are not sure what the SID is, you can obtain it from the INSTANCE_NAME parameter.

The table below describes the `SERVICE_NAMES` and `INSTANCE_NAME` parameters.

Parameter	Description
<code>SERVICE_NAMES</code>	One or more comma separated strings that represent the names of the services on the network. It is possible to provide multiple services names (by individual <code>SERVICE_NAMES</code> entries) so that different usages of a instance can be identified separately. Service names can also be used to identify a service that is available from multiple instances through the use of replication.
<code>INSTANCE_NAME</code>	A string value representing the name of instance and is used to uniquely identify a specific instance when multiple instances share common services names. <code>INSTANCE_NAME</code> should not be confused with the <code>SID</code> , which actually uniquely identifies the instances shared memory on a host.

Configuration Models

Net8 configuration is based upon one of three models:

Network Configuration Model	Suitable When...	Naming Method
Localized management	Network addresses are mapped in a <code>TNSNAMES.ORA</code> file on each node (no Oracle Names server).	Host Naming Local Naming External Naming
Centralized management	Networks where an Oracle Names server performs network address resolution. An Oracle Names server stores client configuration profiles in one location.	Oracle Names

Naming Methods

Net8 can resolve net service names using the following methods:

Naming Method	Description
Host Naming	<p>Instead of net service names, clients use an alias that is mapped to a database's global database name during a connection. A host name is used to identify a database's global database name via an existing name resolution service. This name resolution service might be Domain Name Services (DNS), Network Information Service (NIS) or simply a centrally-maintained set of <code>/etc/hosts</code> files. Host naming allows users to connect to an Oracle server by simply providing the server computer's host name or host name alias. No client configuration is required to take advantage of this feature. This method is recommended for simple TCP/IP environments.</p> <p>Additional Information: See "Configuring the Network with the Host Naming Method" on page 6-7.</p>
Local Naming	<p>Resolves net service names to network addresses by using information configured and stored on each individual client. Local naming is most appropriate for simple distributed networks with a small number of services that change infrequently.</p> <p>Additional Information: See "Configuring the Network with the Local Naming Method" on page 6-12.</p>
Oracle Names	<p>Oracle Names stores names and addresses of all services on a network on an Oracle Names server. Connection requests are routed through an Oracle Names server, which resolves the net service name to a network address. The information is then returned to the client.</p> <p>Additional Information: See "Configuring the Network with the Oracle Names Method" on page 6-21.</p>
External Naming	<p>External naming refers to net service name resolution by using a supported third-party naming service.</p> <p>Additional Information: See "Configuring the Network with the External Naming Method" on page 6-51.</p>

For further information regarding choosing the right method, see "[Resolving Net Service Names](#)" on page 3-3.

Configuration Files

Depending on the naming method you use, the following configuration files may be created or modified in `$ORACLE_HOME/network/admin` on UNIX and `ORACLE_HOME\network\admin` on Windows platforms:

Configuration File	Description
LISTENER.ORA	Located on the server, this file includes addresses of all listeners on a server, and various control parameters used by the listener.
NAMES.ORA	Located on the Oracle Names server, this file includes the location, domain information, and optional configuration parameters for each Names server.
SQLNET.ORA	Includes the names resolution method. This file resides on clients.
TNSNAMES.ORA	Located on the clients, this file includes a list of net service names and their listener address and service descriptions.

See also the following:

- [Appendix B, "Sample Configuration Files"](#) for samples of these files
- [Appendix C, "Configuration Parameters"](#) for parameter information

Configuration of these files is described for each naming method in this chapter.

Configuring the Network with the Host Naming Method

Note: This section describes configuration from a release 8.1 perspective. If you have an existing release 8.0 or release 7.x configuration, see [Chapter 4, "Understanding Release 8.1 Installation and Net8 Products"](#).

Host naming can eliminate the need for service name lookup in the TNSNAMES.ORA configuration files in environments where simple connectivity is desired. However, host naming is not suitable for large or complex environments where advanced features such as connection pooling, external procedures, or heterogeneous services, which require additional connect information, are desired. In these cases, another naming method is recommended.

The host naming method is available for TCP/IP network environments only. When instances register with the listener, the global database name of the server, a name comprised of the database name and domain name, is registered. The global database name is mapped to host name and alias in an existing name resolution service. Client use the alias in their connect string to connect to the database.

Clients may connect to a server using the alias if:

- you are connecting to an Oracle8i database service with Net8 Server/Net8 Client software installed
- your client and server are connecting over a TCP/IP protocol
- all names are resolved through an IP address translation mechanism such as Domain Name Service (DNS), or a centrally maintained TCP/IP hosts file
- no advanced features like Oracle Connection Manager or security are requested or required

To configure the host naming method, perform these tasks:

[Step 1: Configure the Listener with the Global Database Name](#)

[Step 2: Configure HOSTNAME as the First Naming Method](#)

[Step 3: Set Up Host Name Resolution Environment](#)

Step 1: Configure the Listener with the Global Database Name

The global database name must be registered with the listener. How this registration takes place depends upon the release of the database.

Oracle8i release 8.1

If the database can find the listener, information about the database is automatically registered with the listener, including the global database name. The listener is found if:

- the default listener named LISTENER running on TCP/IP on port 1521 is running
- the LOCAL_LISTENER parameter is set in the INITSID.ORA file

If the database cannot find the listener, the LISTENER.ORA file must be configured with the GLOBAL_DBNAME parameter, as described in "[Oracle8i Release 8.0 or Previous](#)" below.

Oracle8i Release 8.0 or Previous

If the database is an Oracle8i release 8.0 database or Oracle7 database, database information is registered with the listener through the LISTENER.ORA file. For these database, statically configured the SID_LIST_listener_name section to include the GLOBAL_DBNAME parameter, as shown below:

```
sid_list_listener=(sid_list=  
(sid_desc=  
(global_dbname=sales.com)  
(sid_name=db1)  
(oracle_home=/u01/app/oracle/8.0.5)  
)
```

See "[Configuring a Listener with Service Information](#)" on page 6-71 for further information.

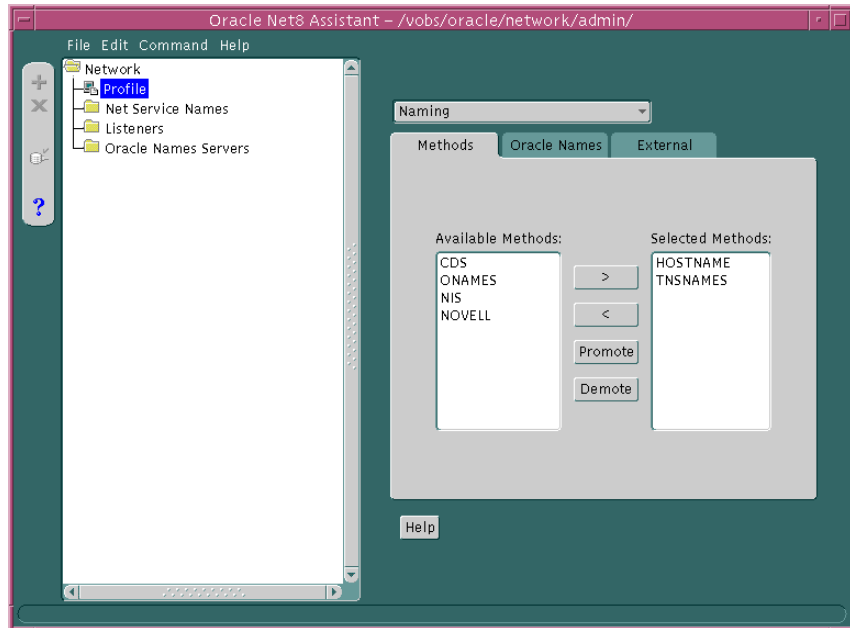
Step 2: Configure HOSTNAME as the First Naming Method

Configure host naming as the first method specified in the NAMES_DIRECTORY_PATH parameter in the SQLNET.ORA file. This parameter specifies the order of naming methods Net8 will use to reconcile services.

To specify local naming as the first naming method:

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Click the Profile icon.
3. Choose Naming from the drop-down list box.
4. Click the Methods tab.
5. Choose TNSNAMES from the Available Methods list, then click ">".

- e. Select TNSNAMES in the Selected Methods list, then use the Promote button to move the selection to the top of the list:



- f. Choose Save Network Configuration from the File menu.
- g. Choose Exit from the File menu to exit the Net8 Assistant application.

The SQLNET.ORA file should contain an entry that lists HOSTNAME first in the NAMES.DIRECTORY_PATH parameter:

```
names.directory_path=(hostname, tnsnames)
```

Step 3: Set Up Host Name Resolution Environment

The global database name must be resolved through an IP address translation mechanism, such as DNS, NIS, or a centrally-maintained TCP/IP host file, `/etc/hosts`. The alias created for the global database name must match the global database name.

It is important to note that the domain portion of the global database name must match the network domain.

For example, if a global database name of SALES.COM for a database exists on a machine named SALES-PC, the entry in the `/etc/hosts` file would like the following.

```
#IP address of server    host name  alias
144.25.219.146         sales-pc  sales.com
```

Clients would use SALES.COM in their connect string, as described in ["Connecting to a Database"](#) on page 8-2. If the client and server are in the same domain of COM, the client only needs to enter SALES in the connect string.

Configuring the Network with the Local Naming Method

Note: This section describes configuration from a release 8.1 perspective. If you have an existing release 8.0 or release 7.x configuration, see [Chapter 4, "Understanding Release 8.1 Installation and Net8 Products"](#).

With the local naming method, net service names are added to the TNSNAMES.ORA file. A net service name is mapped to a network address contained in a connect descriptor. A *connect descriptor* contains the location of the listener through a protocol address and the service name of the service to which to connect. Clients use this net service name when making a connection with an application.

An example of a TNSNAMES.ORA file's basic syntax for a net service name mapped to a connect descriptor is shown in [Figure 6-1](#). DESCRIPTION contains the connect descriptor, ADDRESS is the listener address, and CONNECT_DATA contains the service name.

Figure 6-1 Net Service Name Usage in TNSNAMES.ORA

```
net_service_name=  
(description=  
  (address= (protocol_address_information))  
  (connect_data=  
    (service_name=service_name)  
  )  
)  
)
```

Each connect descriptor can contain one or more listener addresses, as shown in [Figure 6-2](#):

Figure 6-2 Multiple Listener Addresses

```
net_service_name=
(description=
  (address= (protocol_address_information))
  (address= (protocol_address_information))
  (address= (protocol_address_information))
  (connect_data=
    (service_name=service_name)
  )
)
```

Note: If there was only one list of addresses, it was required to use an ADDRESS_LIST.

Optional parameters, such as connect-time failover, client load balancing, and Oracle Connection Manager features, may be set for a list of addresses within a connect descriptor or all addresses among multiple connect descriptors.

For more information about these features, see [Chapter 7, "Enabling Optional Net8 Features"](#) and ["Local Naming Parameters \(TNSNAMES.ORA\)"](#) on page C-31.

Configuration During Installation

After a Custom installation on the client or server, the Net8 Configuration Assistant prompts you to configure a naming methods. If the local naming method is chosen, the Net8 Configuration Assistants prompts you to configure net service names in the TNSNAMES.ORA file. If you choose to use the Net8 Configuration Assistant to create net service names, ensure the listener protocol addresses match in the TNSNAMES.ORA and LISTENER.ORA files. Use the online help for instruction.

Configuration After Installation

Net service names can be added to the TNSNAMES.ORA file at any time. To configure the local naming method, perform these tasks:

[Step 1: Configure Net Service Names](#)

[Step 2: Configure TNSNAMES as the First Naming Method](#)

[Step 3: Distribute Configuration](#)

[Step 4: Configure the Listener](#)

Step 1: Configure Net Service Names

To configure with the local naming method, use either the Net8 Assistant or the Net8 Configuration Assistant on the server:

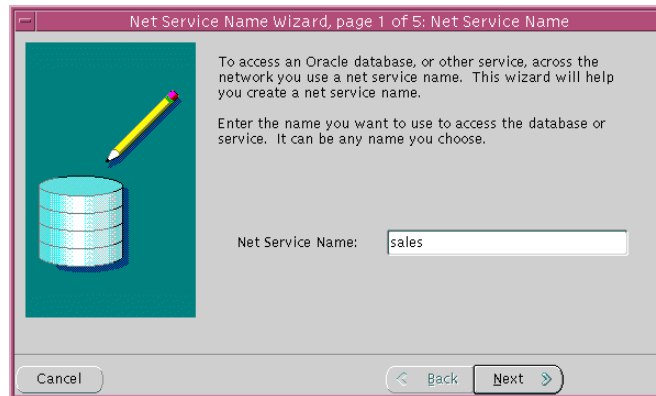
Note: Net service names may also be created by running Net8 Easy Config. In the future this tool may not be available. Therefore, Oracle Corporation recommends using either the Net8 Configuration Assistant or the Net8 Assistant. To run Net8 Easy Config:

- On UNIX, run `netec` at `$ORACLE_HOME/bin`.
 - On Windows platforms, choose Start > Programs > Oracle - `HOME_NAME` > Network Administration > Net8 Configuration Assistant.
-
-

Net8 Assistant To use the Net8 Assistant:

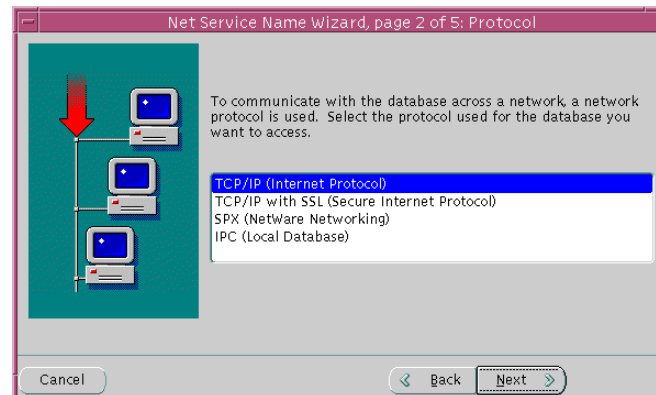
1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - `HOME_NAME` > Network Administration > Net8 Assistant.
2. Click the Net Service Names folder in the directory tree. If net service names exist, they display in the tree structure.
3. Select Create from the Edit menu or click on the "+" button to create a new net service name. The *Net Service Name Wizard* starts.

4. Enter any name in the Net Service Name field, then click Next:

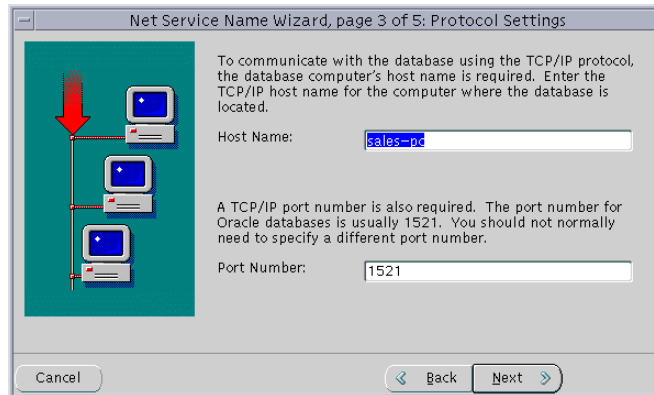


If you want to be explicit, the net service name may be qualified with the client's domain. The net service name is automatically domain qualified if the `NAMES.DEFAULT_DOMAIN` parameter in `SQLNET.ORA` is specified. For further information, see "[Configuring Domains](#)" on page 7-16.

5. Select the protocol the listener is configured to listen on (this protocol must also be installed on the clients), then click Next:



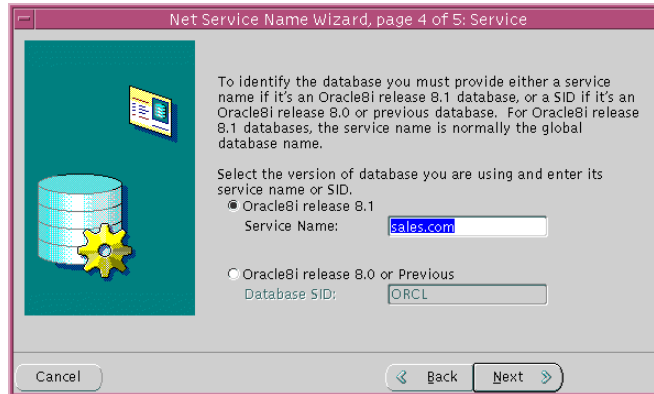
6. Enter the appropriate listener protocol address information for your chosen protocol in the fields, then click Next:



See "[Configuring Protocol Addresses](#)" on page 6-76 for further information about protocol addresses.

When configuring the listener to listen on TCP/IP for typical Net8 clients, you should choose the default port or 1521. If you do not, you must configure the LOCAL_LISTENER parameter in the INITSID.ORA file and resolve the listener name through a naming method. For further information about configuring non-default port numbers in LISTENER.ORA, see "[Configuring a Listener that Uses a Non-Default Port](#)" on page 6-67.

7. If the destination service is release 8.1, click *Oracle8i release 8.1*, enter a service name in the Service Name field, then click Next:

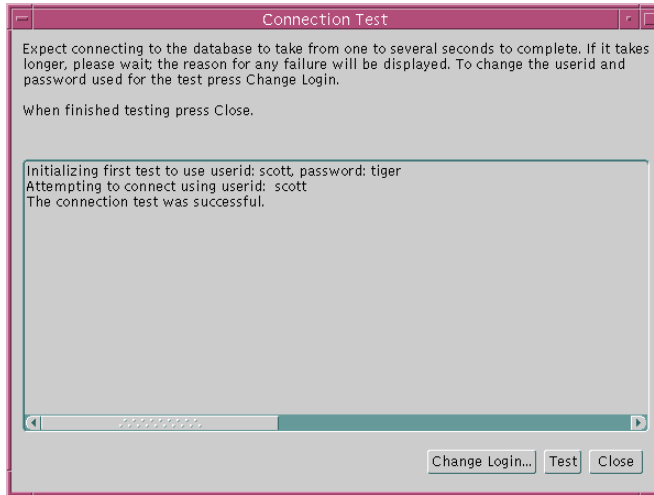


The service name is typically the global database name. See "[Setting Service Names and Instance Names](#)" on page 6-3 for further information about the service name string to use.

If destination service is prior to release 8.1, click Oracle8 Version 8.0 or Previous, enter a SID in the Database SID field, then click Next.

8. Click Test to verify the net service name will work, then click Next. Testing assumes the database and listener are running. If they are not, see "[Net8 Component Startup Overview](#)" on page 8-2 to start components, or click Next to disregard testing and continue to step 10.

A successful test results in "*The connect test was successful.*" message in the *Connect Test* dialog box:



If the test was not successful:

- Ensure the database and listener are running, then click Test.
 - Click Change Login to change the user name and password for the connection, then click Test.
9. Click Close to dismiss the *Connect Test* dialog box.
 10. Click Finish to save your configuration and dismiss *Net Service Names Wizard*.
 11. Choose Save Network Configuration from the File menu.
 12. Choose Exit from the File menu to exit the Net8 Assistant.

See also the following:

- ["Configuring Advanced Connection Attributes for a Net Service Name"](#) on page 7-7 and ["Configuring Multiple Address Options"](#) on page 7-11 to set optional address and multiple address characteristics.
- ["Configuring Domains"](#) on page 7-16 to specify a domain to request net service names.

Net8 Configuration Assistant To use the Net8 Configuration Assistant:

1. Start the Net8 Configuration Assistant:

- On UNIX, run `netca` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Configuration Assistant.
2. Select *Local Net8 Service Name configuration*, then click Next.
 3. Click Create.
 4. Follow the prompts in the wizard and online help to complete net service name creation.

Note: If you select the *Oracle8i release 8.1 database or service* option, the service name you enter is typically the global database name. See "[Setting Service Names and Instance Names](#)" on page 6-3 for further information about the service name string to use.

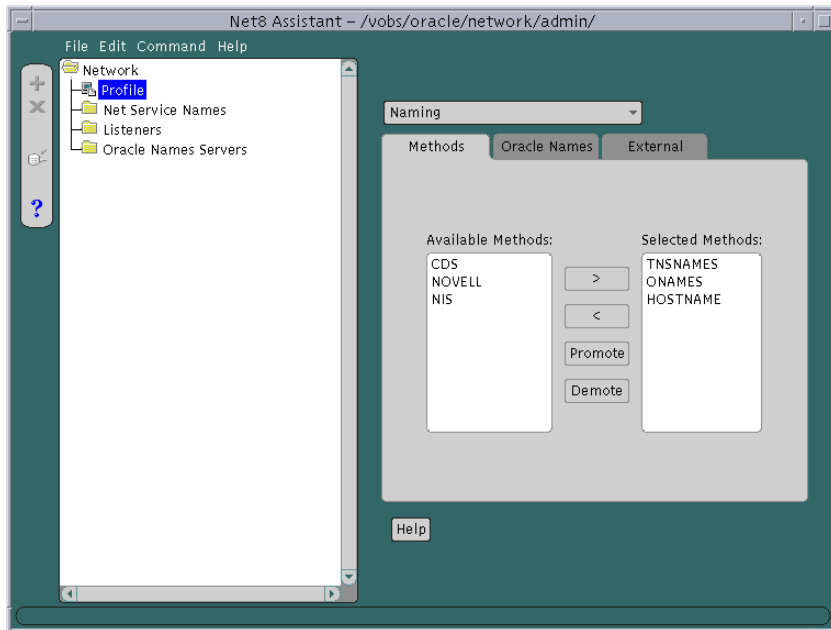
Step 2: Configure TNSNAMES as the First Naming Method

Configure local naming as the first method specified in the `NAMES_DIRECTORY_PATH` parameter in the `SQLNET.ORA` file. This parameter specifies the order of naming methods Net8 will use to reconcile services.

To specify local naming as the first naming method:

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Click the Profile icon.
3. Choose Naming from the drop-down list box.
4. Click the Methods tab.
5. Choose TNSNAMES from the Available Methods list, then click ">".

- h. Select TNSNAMES in the Selected Methods list, then use the Promote button to move the selection to the top of the list:



- i. Choose Save Network Configuration from the File menu.
- j. Choose Exit from the File menu to exit the Net8 Assistant application.

The SQLNET.ORA file should contain an entry that lists TNSNAMES first in the NAMES.DIRECTORY_PATH parameter:

```
names.directory_path=(tnsnames, onames, hostname)
```

Step 3: Distribute Configuration

After one client is configured, it is best to simply copy over the TNSNAMES.ORA and SQLNET.ORA configuration files to the same location on the clients. This will ensure the files are consistent. Otherwise, you must use Net8 Assistant or Net8 Configuration Assistant on every client, introducing possible errors.

Step 4: Configure the Listener

Ensure the listener (located on the server) is configured to "listen on" on the same protocol and protocol address you configured for the net service name. By default, the listener should already be configured for your operating system's primary protocol, typically TCP/IP. For further information about configuring the listener, see "[Configuring Protocol Addresses](#)" on page 6-76.

Configuring the Network with the Oracle Names Method

Note: This section describes configuration from a release 8.1 perspective. If you have an existing release 8.0 or release 7.x configuration, see [Chapter 4, "Understanding Release 8.1 Installation and Net8 Products"](#).

Oracle Names simplifies the setup and administration of global, client/server computing networks. Oracle Names makes network address and database link information available to all nodes throughout the network. Each database server's network address is identified with a simple service name. Client applications then can request a database connection with that name rather than a lengthy address. Oracle Names shields users and applications from changes made to the network infrastructure. It provides for centralized administration of network service names.

An Oracle Names server stores the following information:

Data	Description
database global database names and addresses	The Oracle Names server retrieves information about the database, including the global database name (database name and domain) and address, from the listener. The address is configured in the LISTENER.ORA file, and the global database name is registered during database startup or statically configured in the LISTENER.ORA file. You do not to register this information.
other Oracle Names server names and addresses	An Oracle Names server stores the names and addresses of all other Oracle Names servers in the same administrative region. If there is more than one administrative region in a network, the Oracle Names server will store the name and address of at least one Oracle Names server in the root administrative region and each of the immediate sub-regions. You do not need to register this information.

Data	Description
net service names	<p>If you register net service name with the NAMESCTL control utility or the Net8 Assistant, an Oracle Names server stores them. An Oracle Names server also stores gateways to non-Oracle databases and Oracle RDB databases.</p>
global database link	<p><i>Database links</i> allow a database to communicate with another database another. The name of a database link is the same as the global database name of the database to which the link points. Typically, only one database link should exist per database.</p> <p>The following types of database links may be created:</p> <ul style="list-style-type: none"> ■ You can create a <i>private database link</i> in a specific schema of a database. Only the owner of a private database link can use it. ■ You can create a <i>public database link</i> for a database. All users and in the database can use it. ■ You can create a <i>global database link</i> in Oracle Names server so anyone in the network can use it. <p>Because the Oracle Names retrieves the global database name from the listener, a global database link that is the global database name is automatically registered with the Oracle Names server. Therefore, you do not need to register this information.</p> <p>User name and password credentials for the global database link may be registered with the Oracle Names server using the Net8 Assistant. These global database links may be supplemented with link qualifiers defined through the Net8 Assistant.</p> <p>Global database links may be superseded with private and public database links created by individual users. For more information about private and public database links, see <i>Oracle8i Distributed Database Systems</i>.</p>
aliases	<p>An Oracle Names server stores aliases or alternative service names for any defined net service name, database service or global database link. Aliases may be registered with the Oracle Names server using either the NAMESCTL control utility or Net8 Assistant.</p>
Oracle Connection Managers	<p>An Oracle Names server stores the names and listening addresses of all Oracle Connection Managers on the network. You do not to register this information.</p>

Configuring Oracle Names involves the following tasks:

[Step 1: Consider Oracle Names Options](#)

[Step 2: Install](#)

[Step 3: Create an Oracle Names Server](#)

[Step 4: Configure Clients and Database Servers To Use Oracle Names Servers](#)

[Step 5: Configure the Listener](#)

[Step 6: Register Data with the Oracle Names Server](#)

[Step 7: \(Optional\) Delegate Domains to Delegated Administrative Regions](#)

Step 1: Consider Oracle Names Options

Prior to creating an Oracle Names server, you must decide whether:

- You need service registration data replicated continuously among Oracle Names servers, or want all Oracle Names servers within a region to store their registration data in an Oracle database.
- You need support for one or more *administrative regions*. An administrative region is a collection of Oracle Names servers in one or more *domains*, a grouping of network objects, such as databases. Networks with multiple administrative regions must have one *root administrative region* and one or more *delegated administrative regions*. Delegated administrative regions contain the domains and Oracle Names server addresses in any alternate regions which act as direct child regions of the root.

For more assistance with the decisions above, see "[Oracle Names Architecture](#)" on page 2-31.

Step 2: Install

Ensure the following is installed:

- Oracle Names on its own machine that is designated as the Oracle Names server
- Net8 Client or SQL*Net Client on the clients
- Net8 Server or SQL*Net Server on the server

Important: The Oracle Universal Installer does not install the `$ORACLE_HOME/network/names` directory on UNIX platforms. This directory is necessary for successful configuration. If it does not exist, manually create it.

Step 3: Create an Oracle Names Server

How an Oracle Names server is created depends upon whether you want region data:

- stored in tables, called a region database, in an Oracle database accessible to Oracle Names servers

When information is added to an Oracle Names server, the information is stored in the database and in the Oracle Names server's cache files. If the database is unavailable, the information in the cache files is used.

- stored in an Oracle Names server's cache files and instantly replicated to the cache files of all other Oracle Names servers

The cache files are stored in `$ORACLE_HOME/network/names` on UNIX platforms and `ORACLE_HOME\network\names` on Windows NT:

File	Description
CKCFG.ORA	Contains information about the Oracle Names server configuration, similar to the NAMES.ORA file.
CKPCCH.ORA	Contains client cache information, such as the connect string information.
CKPREG.ORA	Contains information about data stored in the Oracle Names server, which is similar to a TNSNAMES.ORA file.

A default Oracle Names server named `ONAMES_host` is created during installation for a non-region database configuration. This Oracle Names is configured to listener on TCP/IP with port 1575. It contains basic configuration that does not include multiple domains or multiple regions. If you would like to use this Oracle Names server, skip this step and proceed to "[Step 2: Discover Oracle Names Servers](#)" on page 6-36.

The following sections cover both modes:

- [Store Data in a Region Database](#)
- [Store Data in Files](#)

Whichever method you choose, Oracle Corporation recommends you create more than one Oracle Names server for the network, in case one should go down.

Store Data in a Region Database

To store service data in an Oracle database, perform the following tasks:

- [Step 1: Configure the Database Server](#)
- [Step 2: Create Oracle Names Servers](#)
- [Step 3: Start the Oracle Names Server](#)

Step 1: Configure the Database Server To configure the database server:

1. Start the database if it is not currently running. Otherwise, go to step 2.

- a. Connect to the database as INTERNAL:

```
sql> CONNECT internal/password
```

where *password* is ORACLE for the INTERNAL user account by default.

- b. Start the database:

```
sql> STARTUP;
```

2. Connect to the database as the SYSTEM user:

```
sql> CONNECT system/password
```

where *password* is MANAGER for the SYSTEM user account by default.

3. Create a user account and password:

```
sql> CREATE user user  
       identified by password  
       default tablespace users  
       temporary tablespace temp;
```

4. Run the NAMESINI.SQL script located in \$ORACLE_HOME/network/admin on UNIX and ORACLE_HOME\network\admin\names on Windows NT platforms. This script creates the tables needed by Oracle Names to store information. Optionally, run NAMESUPG.SQL to define old tables.

```
sql> CONNECT user/password  
sql> @oracle_home/network/admin/namesini.sql;
```

Step 2: Create Oracle Names Servers For each machine where Oracle Names is installed and where you want an Oracle Names server, create an Oracle Names server from the Net8 Assistant.

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. From the Net8 Assistant, click the Oracle Names Servers folder.
3. Choose Create from the Edit menu, or click on the "+" button to create a new Oracle Names server.

The *Names Wizard* starts.

The wizard guides you through the creation and configuration process, prompting for:

- a unique Oracle Name Server name
- a protocol address for the Oracle Names server
 - If you choose TCP/IP, Oracle Corporation recommends using the default and officially registered port of 1575 for the Oracle Names server.
- a choice to store information in a database or replicate information among Oracle Names servers
 - (Click *Use a region database*.)
- a protocol address for a database's listener
 - If you choose TCP/IP, Oracle Corporation recommends using the default and officially registered port of 1521 for the listener.
- database user ID, password, and service name or SID
- whether or not this Oracle Names server is in the root administrative region

If you specify this Oracle Names server is in the root administrative region, configuration completes.

If you specify this Oracle Names server is not in the root administrative region, it assumed this Oracle Names server is in a delegated administrative region. The wizard then prompts you for the local administrative region's domain name and the address of an Oracle Names server in the root administrative region before completing.

When the wizard completes, the following message appears:

A Names Server, onames_server, has been created with default settings. Use the Configure Server section to modify the default configuration.

4. Select Save Network Configuration from the File menu.
5. Select Exit from the File menu to exit the Net8 Assistant application.
6. Repeat steps 2-5 to create additional Oracle Names servers in a region. The Net8 Assistant does not support creation of multiple Oracle Names servers on one machine.

The Net8 Assistant creates a NAMES.ORA file with at least the following settings:

- Oracle Names server name (NAMES.SERVER_NAME)
- Oracle Names server listening protocol address (NAMES.ADDRESSES)
- database information (NAMES.ADMIN_REGION)

A NAMES.ORA file is shown below with annotations of content:

```
#Oracle Names server name. The name should include the name of the domain
this Oracle Names server is in.
names.server_name=namesvr2.com

#Oracle Names server listening protocol address
names.addresses=
  (address=(protocol=tcp)(host=namesrv2-pc)(port=1575))
  [(address=...)]

#Database repository information
names.admin_region=
  (region=
    (description=
      (address=(protocol=tcp)(host=sales-pc)(port=1521))
      (connect_data=
        (service_name=sales.com)
      )
    )
    (userid=system)
    (password=password)
    (name=local_region)
    (refresh=86400)
    (retry=60)
    (expire=600)
  )
```

```
#If an Oracle Names server is in a delegated administrative region, identify
the address of an Oracle Names server in the root administrative region.
names.domain_hints=
  (address=(protocol=tcp)(host=namesrv1-pc)(port=1575))
```

```
# Specify the domain controlled by this region and the time to live (TTL).
If this is the root administrative region, you must have NAME= (null) to
identify the root domain properly.
names.domains=
  (domain=
    (name=)
    (min_ttl=86400)
  )
```

See "[Configuring Protocol Addresses](#)" on page 6-76 for more information about protocol syntax needed for address information.

Administering Multiple Domains

If you want the region to administer more than one domain, specify the additional domains in the NAMES.DOMAINS parameter with the Net8 Assistant:

1. Double-click on the Oracle Names server folder.
2. Select the Oracle Names server.
3. Select Domains from the drop-down list box.
4. Select the Domains tab.
5. Enter the domain name in the Domain Name field.
6. Enter the time-to-live information.
7. Click Add.
8. Select Save Network Configuration from the File menu.
9. Repeat steps 1-8 to create additional for additional domains.

In the example below, NAMES.DOMAINS contains a listing for the root, COM, ORACLE.COM, and HQ.ORACLE.COM domains. All the domain precedent to HQ.ORACLE.COM must be defined in order to define HQ.ORACLE.COM.

```
names.domains=
(domain_list=
  (domain=
    (name=)
    (min_ttl=86400)
  )
  (domain=
    (name=com)
    (min_ttl=86400)
  )
  (domain=
    (name=oracle.com)
    (min_ttl=86400)
  )
  (domain=
    (name=hq.oracle.com)
    (min_ttl=86400)
  )
)
```


Step 3: Start the Oracle Names Server Start the Oracle Names server using either the Net8 Assistant or the NAMESCTL control utility:

Use the Net8 Assistant...	Use the control utility NAMESCTL...
<ol style="list-style-type: none"> 1. Double-click on the Oracle Names server folder. 2. Select the Oracle Names server. 3. Select Manage Server from the drop-down list box. 4. Select the Control tab. 5. Click the Start radio button from the Server Operations field. 6. Click Apply. <p>The following message appears: <i>Server started successfully.</i></p> <p>On Windows NT, a service called OracleHOME_NAMENamesONAMES_SERVER is created.</p>	<p>From the command line, enter:</p> <pre>namesctl</pre> <p>If the following error messages appear, ignore them.</p> <pre>NNL-00024: warning: no preferred names servers in SQLNET.ORA >NNL-00018: warning: could not contact default name server</pre> <pre>namesctl> start</pre> <p>The START command of NAMESCTL loads the Oracle Names server into memory and tells it to begin executing. At startup, the Oracle Names server loads its configuration and data.</p> <p>On Windows NT, this creates a service called OracleORACLE_HOMENamesonames_serverService.</p>

Store Data in Files

To store data in files and have it replicated between Oracle Names server during updates, performs these tasks:

- [Step 1: Create an Oracle Names Server](#)
- [Step 2: Discover Oracle Names Servers](#)

Step 1: Create an Oracle Names Server For each machine where Oracle Names is installed and where you want an Oracle Names server, create an Oracle Names server from the Net8 Assistant:

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. From the Net8 Assistant, click the Oracle Names Servers folder.
3. Choose Create from the Edit menu, or click on the "+" button to create a new Oracle Names server.

The *Names Wizard* starts.

The wizard guides you through the creation and configuration process, prompting you for

- a unique Oracle Name Server name
- a listening protocol address for the Oracle Names server
If you choose TCP/IP, Oracle Corporation recommends using the default and officially registered port of 1575 for the Oracle Names server.
- a choice to store information in a database or replicate information among Oracle Names servers
(Click *Don't use a region database*.)
- to identify if this is the first Oracle Names server in the region
If this is not the first Oracle Names server in the region, the wizard then prompts you to discover the other Oracle Names servers or to specify the address of another Oracle Names server in the region.
- to identify if this Oracle Names server is in the root administrative region

If you specify this Oracle Names server is in the root administrative region, configuration completes.

If you specify this Oracle Names server is not in the root administrative region, it assumed this Oracle Names server is in a delegated administrative region. The wizard then prompts you for the local administrative region's domain name and the address of an Oracle Names server in the root administrative region before completing.

When the wizard completes, the following message appears:

A Names Server, onames_server, has been created with default settings. Use the Configure Server section to modify the default configuration.

4. Select Save Network Configuration from the File menu.
5. Repeat Steps 2-4 to create additional Oracle Names servers in a region. The Net8 Assistant does not support creation of multiple Oracle Names servers on one machine.

The Net8 Assistant creates a NAMES.ORA file with at least the following settings:

- Oracle Names server name (NAMES.SERVER_NAME)
- Oracle Names listening protocol address (NAMES.ADDRESSES)

A NAMES.ORA file is shown below with annotations of content:

```
#Oracle Names server name. The name should include the name of the domain
this Oracle Names server is in.
```

```
names.server_name=namesvr2.com
```

```
#Oracle Names server listening protocol address
```

```
(address=(protocol=tcp)(host=namesrv2-pc)(port=1575))
```

```
[(address=...)]
```

```
#If an Oracle Names server is in a delegated administrative region, identify
the address of an Oracle Names server in the root administrative region.
```

```
names.domain_hints=
```

```
(address=(protocol=tcp)(host=namesrv1-pc)(port=1575))
```

```
# Specify the domain controlled by this region and the time to live (TTL).
If this is the root administrative region, you must have NAME= (null) to
identify the root domain properly.
```

```
names.domains=
```

```
(domain=
```

```
(name=)
```

```
(min_ttl=86400)
```

```
)
```

See "[Configuring Protocol Addresses](#)" on page 6-76 for more information about protocol syntax needed for address information.

Administering Multiple Domains

If you want the region to administer more than one domain, specify the additional domains in the NAMES.DOMAINS parameter with the Net8 Assistant:

1. Double-click on the Oracle Names server folder.
2. Select the Oracle Names server.
3. Select Domains from the drop-down list box.
4. Select the Domains tab.
5. Enter the domain name in the Domain Name field.
6. Enter the time-to-live information.
7. Click Add.
8. Select Save Network Configuration from the File menu.
9. Repeat steps 1-8 to create additional for additional domains.

In the example below, NAMES.DOMAINS contains a listing for the root, COM, ORACLE.COM, and HQ.ORACLE.COM domains. All the domain precedent to HQ.ORACLE.COM must be defined in order to define HQ.ORACLE.COM.

```
names.domains=
  (domain_list=
    (domain=
      (name=)
      (min_ttl=86400)
    )
    (domain=
      (name=com)
      (min_ttl=86400)
    )
    (domain=
      (name=oracle.com)
      (min_ttl=86400)
    )
    (domain=
      (name=hq.oracle.com)
      (min_ttl=86400)
    )
  )
)
```

Step 2: Discover Oracle Names Servers After all the Oracle Names servers are configured, have each Oracle Names, except the first Oracle Names server in the region, discover the other Oracle Names servers in a region.

See "[Understanding Discovery](#)" on page 2-45 for a description of discovery.

On the machine where Oracle Names resides, use either the Net8 Assistant or the NAMESCTL control utility:

Use the Net8 Assistant...

1. Double-click on the Oracle Names Server folder.
2. Select an Oracle Names server.
3. Choose Manage Server from the drop-down list box.
4. Click the Control tab.
5. Click the Start option button from the Server Operations field.
6. Click Apply.

The following message appears:

Server started successfully.

7. Choose Discover Oracle Names Servers from the Command menu.

This produces an .SDNS.ORA file in \$ORACLE_HOME/network/names on UNIX or SDNS.ORA in ORACLE_HOME\network\names on Windows platforms. This file contains the names and addresses of the Oracle Names server(s). This file is read to find the addresses of Oracle Names servers.

The following message appears:

"Discovered Oracle Names Server in the region. Please exit the tool and start again."

If an Oracle Names server does not respond, and you know the network address of a particular Oracle Names server, a dialog will prompt you for that address.

Use the control utility NAMESCTL...

1. Start the Oracle Names server:

```
namesctl
namesctl> start
```

2. Discover the other Oracle Names servers:

```
namesctl> reorder_ns
```

This produces an .SDNS.ORA file in \$ORACLE_HOME/network/names on UNIX or SDNS.ORA in ORACLE_HOME\network\names on Windows platforms. This file contains the names and addresses of the Oracle Names server(s). This file is read to find the addresses of Oracle Names servers.

If an Oracle Names server cannot be found, and you know the network address of a particular Oracle Names server, enter the address at the prompt. For example:

```
namesctl> reorder_ns
(address=(protocol=tcp)(host=mail-pc)(port=1575))
```

3. Query another Oracle Names server to verify discovery.

```
namesctl> query onames_server
```

Step 4: Configure Clients and Database Servers To Use Oracle Names Servers

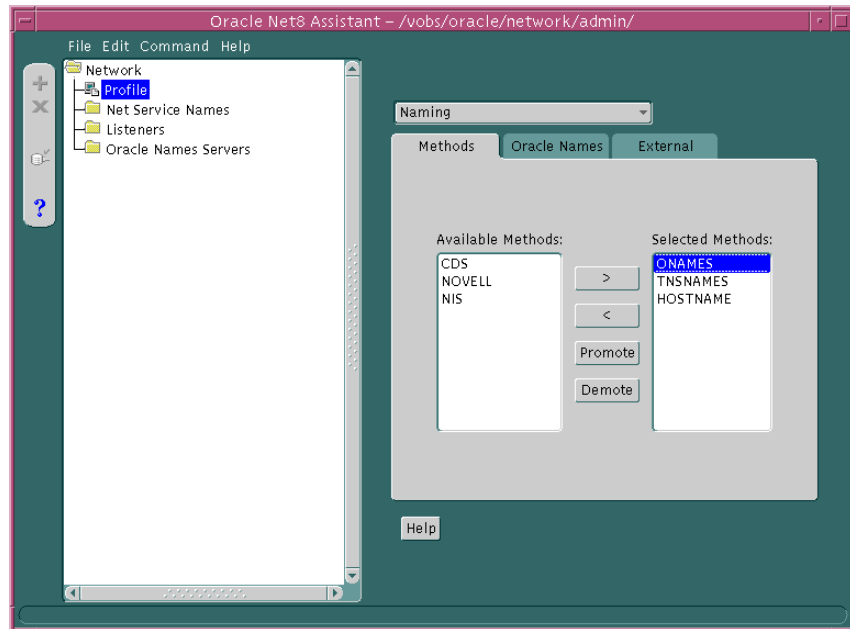
To configure the client and database server machines:

1. Optionally, create a preferred Oracle Names server list, as described in "[Configuring Preferred Oracle Names Servers](#)" on page 7-20.
2. Discover Oracle Names servers by creating an `.SDNS.ORA` file in `$ORACLE_HOME\network\names` on UNIX or `SDNS.ORA` file in `ORACLE_HOME\network\names` on Windows platforms. This file contains the order of names and addresses of the Oracle Names server(s) to contact. Client machines read this file to find the addresses of Oracle Names servers to contact. Use either the Net8 Assistant or the NAMESCTL utility.

See "[Understanding Discovery](#)" on page 2-45 for a description of discovery.

Use the Net8 Assistant...	Use the control utility NAMESCTL...
<ol style="list-style-type: none"> 1. Start the Net8 Assistant: <ul style="list-style-type: none"> -On UNIX, run <code>netasst</code> at <code>\$ORACLE_HOME/bin</code>. -On Windows NT, choose Start > Programs > Oracle - <code>HOME_NAME</code> > Network Administration > Net8 Assistant. 2. Choose Discover Oracle Names Servers from the Command menu. <p>This produces an <code>.SDNS.ORA</code> file in <code>\$ORACLE_HOME/network/names</code> on UNIX or <code>SDNS.ORA</code> <code>ORACLE_HOME\network\names</code> on Windows platforms. This file contains the names and addresses of the Oracle Names server(s). This file is read to find the addresses of Oracle Names servers.</p> <p>The following message appears:</p> <p><i>"Discovered Oracle Names Server in the region. Please exit the tool and start again."</i></p> <p>If an Oracle Names server does not respond, and you know the network address of a particular Oracle Names server, a dialog will prompt you for the that address.</p> 	<p>Discover the other Oracle Names servers:</p> <pre>namesctl namesctl> reorder_ns</pre> <p>This produces an <code>.SDNS.ORA</code> file in <code>\$ORACLE_HOME/network/names</code> on UNIX or <code>SDNS.ORA</code> in <code>ORACLE_HOME\network\names</code> on Windows platforms. This file contains the names and addresses of the Oracle Names server(s). This file is read to find the addresses of Oracle Names servers.</p> <p>If an Oracle Names server cannot be found, and you know the network address of a particular Oracle Names server, enter the address at the prompt. For example:</p> <pre>namesctl> reorder_ns (address=(protocol=tcp)(host=mail-pc)(port=1575))</pre>

3. If you want Oracle Names to be the first method to reconcile net service names, set the NAMES_DIRECTORY_PATH parameter in the SQLNET.ORA file in the following manner:
 - a. Click the Profile icon in the Net8 Assistant.
 - b. Choose Naming from the drop-down list box.
 - c. Click the Methods tab.
 - d. Choose ONAMES from the Available Methods list, then click ">".
 - e. Select ONAMES in the Selected Methods list, then use the Promote button to move the selection to the top of the list:



- f. Choose Save Network Configuration from the File menu.
- g. Choose Exit from the File menu to exit the Net8 Assistant application.

The SQLNET.ORA file should contain an entry that lists ONAMES first in the NAMES_DIRECTORY_PATH parameter:

```
names.directory_path=(onames, tnsnames, hostname)
```


4. Additional features that you may configure for how a client will use Oracle Names include:

Net8 Assistant Option	SQLNET.ORA Parameter	Description
Default Domain	NAMES.DEFAULT_DOMAIN	Indicates the domain name space from which the client will most often request an Oracle Names server. When set, this name will automatically be appended to any unqualified name in an Oracle Names request. Net8 sets the default domain to NULL by default.
Maximum Wait Each Attempt	NAMES.INITIAL_RETRY_TIMEOUT	Specifies how long a client will wait for a response from an Oracle Names server before reiterating the request to another Oracle Names server. Net8 will wait for 15 seconds by default.
Attempts Per Names Server	NAMES.REQUEST_RETRIES	Specifies the number of times a client will attempt to iterate through the list of Oracle Names servers before allowing the operation to fail. Net8 will attempt to iterate through the list of Oracle Names servers once before allowing the operation to fail by default.
Maximum Open Connections	NAMES.MAX_OPEN_CONNECTIONS	Specifies how many connections an Oracle Names client may have open at one time. Net8 allows a client to have 10 connections open at any one time by default. This default value should be sufficient for almost all situations.
Initial Preallocated Requests	NAMES.MESSAGE_POOL_START_SIZE	Allows you to pre-allocate an initial number of messages in a client's message pool. These messages may be used for future requests to Oracle Names servers. Net8 allocates 10 messages in the pool by default. This default value should be sufficient for almost all situations.

To add or configure these features:

- a. Click the Profile icon in the Net8 Assistant.
- b. Choose Naming from the drop-down list box.
- c. Click the Oracle Names tab.
- d. Enter data for the options indicated in the table on the previous page.
- e. Choose Save Network Configuration from the File menu.
- f. Choose Exit from the File menu to exit the Net8 Assistant application.

Step 5: Configure the Listener

Configure the listener with listening database protocol addresses if you are not using your operating system's primary protocol. For further information about configuring the listener, see "[Configuring Protocol Addresses](#)" on page 6-76.

Additionally, set `USE_PLUG_AND_PLAY_listener_name=ON` in the LISTENER.ORA file with the Net8 Assistant. This permits the listener to register database information, such as the service name, SID, and global database name, with Oracle Names servers:

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Double-click on the Listeners folder.
3. Click on a listener.
4. Choose General Parameter from the drop-down list box.
5. Click *Register Service with Oracle Names*.
6. Select Save Network Configuration from the File menu.
7. Choose Exit from the File menu to exit the Net8 Assistant application.

Step 6: Register Data with the Oracle Names Server

As described in the table in "[Configuring the Network with the Oracle Names Method](#)" on page 6-21, Oracle Names server store many kinds of information, including as database locations, Oracle Names server locations, net service names, aliases, global database links, and Oracle Connection Manager. Some of this information is automatically registered with the Oracle Names server because the database registers information with the listener. If you would like to store the following kinds of information, you must register it using either the Net8 Assistant or the NAMESCTL control utility:

- [Net Service Names](#)
- [Global Database Links Credentials and Database Qualifiers](#)
- [Aliases](#)

Registration only needs to occur with *one* Oracle Names server in an administrative region. The other Oracle Names servers will see the information if data is stored in a region database, or the information will be propagated to other Oracle Names if you chose not to use a region database.

After registering these kinds of information, you should query them, as described in "[Testing Network Objects Using NAMESCTL or the Net8 Assistant](#)" on page 8-17.

Net Service Names

Oracle Corporation advises creating a net service name for each service. If a net service name is not created, a user trying to connect will have to pass the global database name to make a connection. Rather than provide this information to your end users, create a net service name for the service.

Use either the Net8 Assistant or the NAMESCTL control utility:

Use the Net8 Assistant...	Use the control utility NAMESCTL...
<ol style="list-style-type: none">1. Start the Net8 Assistant: -On UNIX, run <code>netasst</code> at <code>\$ORACLE_HOME/bin</code>. -On Windows NT, choose Start > Programs > Oracle - <i>HOME_NAME</i> > Network Administration > Net8 Assistant.2. Double-click on the Oracle Names server folder.3. Select an Oracle Names server.4. Choose Manage Data from the drop-down list box.5. Click the Net Service Names tab.6. Select Add.7. Enter an easy-to-remember alias in the Service Name field.8. Choose a protocol and enter the listener address. Additional Information: See "Configuring Protocol Addresses" on page 6-76 for more information about protocol syntax needed for address information.9. Enter the service name for the database, and choose the session type.10. Click Execute.11. Select Save Network Configuration from the File menu.12. Select Exit from the File menu to exit the Net8 Assistant application.	<p>Create and register a net service name with the Oracle Names server:</p> <pre>namesctl namesctl> register net_service_name -D address_data</pre> <p>For example, to register a net service name of SALES with a default TCP/IP listening address, enter:</p> <pre>register sales -D (description=(address=(protocol=tcp)(host=sales-pc)(port=1521)))</pre> <p>Additional Information: See "Configuring Protocol Addresses" on page 6-76 for more information about protocol syntax needed for address information.</p>

If you have been using local naming as your configuration method and want to migrate to an Oracle Names configuration, the net service name information used in the TNSNAMES.ORA file can be uploaded to an Oracle Names server. To load information from a local naming configuration file into an Oracle Names server, use either the Net8 Assistant or the NAMESCTL control utility:

Use the Net8 Assistant...	Use the control utility NAMESCTL...
<ol style="list-style-type: none">1. Select an Oracle Names server.2. Select Manage Data from the drop-down list box.3. Select Load.4. Enter the path for your current master local naming configuration file in the File field.5. Click Execute.6. Select Save Network Configuration from the File menu.7. Select Exit from the File menu to exit the Net8 Assistant application.	<p>From the command line, enter:</p> <pre>namesctl namesctl> load_tnsnames <i>file_name</i></pre>

Global Database Links Credentials and Database Qualifiers

Oracle Corporation recommends creating one global database link per database. A global database link that is the global database name is automatically registered with the Oracle Names server, such as SALES.COM.

For example, a connection to a remote database with the global database link would be:

```
sql> CONNECT @sales.com
```

Therefore, you do not need to register this information unless you want to enter user name and password credentials for it.

To register a global database link, use either the Net8 Assistant or the NAMESCTL control utility. Because the NAMESCTL utility is limited in global database link registration, Oracle Corporation recommends you use the Net8 Assistant.

Use the Net8 Assistant...

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Double-click on the Oracle Names server folder.
3. Select an Oracle Names server.
4. Choose Manage Data from the drop-down list box.
5. Click the DB Links tab.
6. Select Add.
7. Enter the global database link in the DB Link Name field.

NOTE: If the `GLOBAL_NAMES` parameter has been set to `TRUE` in the `INITSID.ORA` file, the name entered must be the global database name.
8. Enter valid user name and password credentials for the database in the User and Password fields.
9. Click Execute.
10. If you want to use a link qualifier, see "[Adding Link Qualifiers](#)" on page 6-46.
11. Select Save Network Configuration from the File menu.
12. Select Exit from the File menu to exit the Net8 Assistant application.

Use the control utility NAMESCTL...

Create and register a global database link with the Oracle Names server:

```
namesctl
namesctl> register dblink -d address_data
```

For example, to register a global database link of SALES.COM with a default TCP/IP listening address, enter:

```
register sales.com -d
(description=(address=(protocol=tcp)(host=sales-pc)(port=1521)))
```

Additional Information: See "[Configuring Protocol Addresses](#)" on page 6-76 for more information about protocol syntax needed for address information.

The NAMESCTL utility will not permit user name and password credentials.

While one global database link per database is recommend, there is no limit on the number of link qualifiers that may be added. Link qualifiers are appended to global database link to provide alternate settings for the database user name and password credentials. For example, a link qualifier of `RDBMS.OMD` can be appended to `SALES.COM`. A connection to a remote database with the global database link and link qualifier would be:

```
sql> CONNECT @sales.com@emp
```

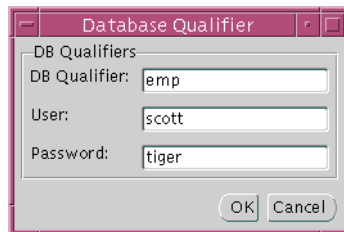
Adding Link Qualifiers

Link qualifiers are primarily for older Oracle database environments where multiple database links are used. Rather than use multiple database links, multiple link qualifiers can be created for a database link.

To register a global link qualifier for a global database link, use the Net8 Assistant. The NAMESCTL utility does not support creation of link qualifiers.

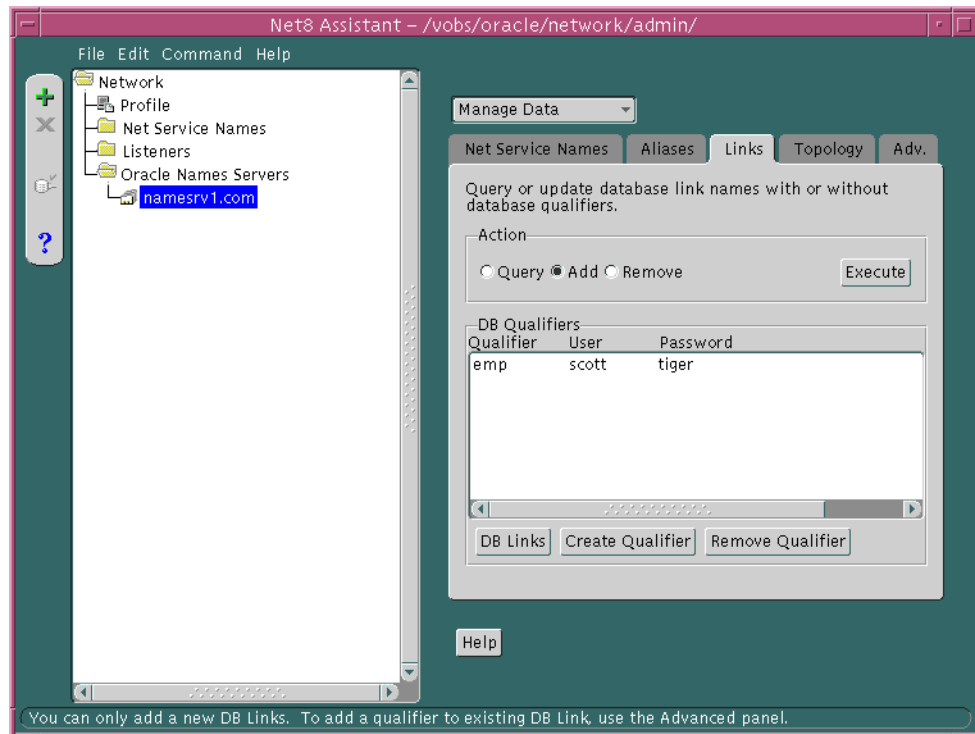
1. Follows steps 1-8 on the procedure on the previous page.
2. Click DB Qualifiers.
3. Click Create Qualifier.

The *Database Qualifier* dialog box appears:



4. Enter a name for the link qualifier, and valid user name and password credentials, then click OK.

The link qualifier is added to the DB Qualifiers list box:



5. Repeat steps 3-5 for each link qualifier you want to add.
6. Click Execute.
7. Select Save Network Configuration from the File menu.
8. Select Exit from the File menu to exit the Net8 Assistant application.

To verify a global database link and link qualifier is working correctly:

1. Connect to the remote database:

```
sql> CONNECT user_name/password@db_link;  
sql> CONNECT user_name/password@db_link@link_qualifier;
```

For example:

```
sql> CONNECT scott/tiger@sales.com;  
sql> CONNECT scott/tiger@sales.com@emp;
```

2. Perform a SELECT statement on a table the user is authorized for:

```
sql> SELECT * from table@db_link;  
sql> SELECT * from table@db_link@link_qualifier;
```

For example:

```
sql> SELECT * from EMP@sales.com;  
sql> SELECT * from EMP@sales.com@emp;
```

For more information about database links, see *Oracle8i Distributed Database Systems*.

Aliases

An alias can be used for type of item stored in the Oracle Names server, such as a Oracle Names server, net service name, or global database link.

To register a global database link, use either the Net8 Assistant or the NAMESCTL control utility.

Use the Net8 Assistant...	Use the control utility NAMESCTL...
<ol style="list-style-type: none"> 1. Start the Net8 Assistant: -On UNIX, run <code>netasst</code> at <code>\$ORACLE_HOME/bin</code>. -On Windows NT, choose Start > Programs > Oracle - <i>HOME_NAME</i> > Network Administration > Net8 Assistant. 2. Double-click on the Oracle Names server folder. 3. Select an Oracle Names server in the administrative region. 4. Choose Manage Data from the drop-down list box. 5. Click the Aliases tab. 6. Select Add. 7. Specify the sub-domain in the Domain Name field. 8. Enter the Oracle Names server name in the Name Server Name field and 9. Click Execute. 10. Select Save Network Configuration from the File menu. 11. Select Exit from the File menu to exit the Net8 Assistant application. 	<p>Create and register an alias with the Oracle Names server:</p> <pre>namesctl namesctl> register alias -d address_data</pre> <p>For example, to register a net service name of SALES with a default TCP/IP listening address, enter:</p> <pre>register sales -d (description=(address=(protocol=tcp)(host=sales-pc)(port=1521)))</pre> <p>Additional Information: See "Configuring Protocol Addresses" on page 6-76 for more information about protocol syntax needed for address information.</p>

Step 7: (Optional) Delegate Domains to Delegated Administrative Regions

If delegated administrative regions are configured, the upper-level administrative region must delegate sub-domains to the delegated administrative regions. Otherwise, the Oracle Names servers in the upper-level administrative region will assume authority over all sub-domains.

Once a domain is delegated, the Oracle Names servers in the current region will forward subsequent operations to the sub-region where the domain is administered by Oracle Names servers.

To delegate a domain, use either the Net8 Assistant or the NAMESCTL control utility.

Use the Net8 Assistant...

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - `HOME_NAME` > Network Administration > Net8 Assistant.
2. Double-click on the Oracle Names server folder.
3. Select an Oracle Names server in the upper-level region.
4. Choose Manage Data from the drop-down list box.
5. Click the Topology tab.
6. Select Delegate Domain.
7. Enter the name of the sub-domain in the Domain Name field.
8. Enter the Oracle Names server that serves the domain in the Names Server Name field, and the listening address for the Oracle Names server.

Additional Information: See "[Configuring Protocol Addresses](#)" on page 6-76 for more information about protocol syntax needed for address information.
9. Click Execute.
10. Select Save Network Configuration from the File menu.
11. Select Exit from the File menu to exit the Net8 Assistant application.

Use the control utility NAMESCTL...

Create and register an alias with the Oracle Names server:

```
namesctl
namesctl> delegate_domain domain_name onames_server
[(description=(address_list=) (address= ...)[])]
```

Additional Information: See "[Configuring Protocol Addresses](#)" on page 6-76 for more information about protocol syntax needed for address information.

For example, to delegate domain ACME.COM serviced by NAMESRV3.ACME.COM, enter:

```
delegate_domain acme.com namesrv3.acme.com
(address=(protocol-tcp) (host=namesrv3-pc) (port=1575))
```

Configuring the Network with the External Naming Method

External naming refers to the method of resolving a net service name to a network address by using a supported non-Oracle naming service. External naming resolve net service names stored in customers' native (non-Oracle) naming services. They include:

- Network Information Service (NIS)
- NetWare Directory Service (NDS)

Configure NIS

Organizations and corporations already using Network Information Service (NIS) as part of their systems infrastructure have the option to store Oracle net service names and addresses in NIS, using NIS External Naming.

When a user gives a command such as

```
sqlplus scott/tiger@payroll
```

(where "payroll" is an Oracle net service name) NIS External Naming on the node running the client program (or server acting as a client program) contacts an NIS server located somewhere in the network, and passes the net service name to the NIS server. The NIS server resolves the net service name into a Net8 address and returns this address to the client program (or server acting as a client program). The client program then uses this address to connect to the Oracle database.

A machine that acts as an NIS server runs a program called `ypserv`, which handles name requests. `ypserv` stores different types of data in special files called *maps*. For example, passwords are stored in a map called `passwd.byname`. Oracle database net service names are stored in a map called `tnsnames`.

When a user issues a command like the one in the previous section, NIS External Naming uses an RPC call to contact the `ypserv` program and passes the Oracle net service name "payroll" and the name of the map—`tnsnames`. The `ypserv` program looks in the `tnsnames` map for the name "payroll" and its corresponding value, which is the address for the net service name. The address is returned to the client, and the client program (or server acting as a client program) uses this address to contact the database server.

System Requirements

NIS External Naming requires SQL*Net 2.2 or greater.

Step 1: Configure NIS Servers to Support the NIS External Naming

Before configuring servers to support the NIS External Naming, make sure that NIS is configured and running on the NIS servers that need to resolve Oracle database net service names. Consult your NIS documentation for specifics.

Add the "tnsnames" Map to the Existing Set of NIS Maps To add the `tnsnames` map to the existing set of NIS maps:

1. Create a `TNSNAMES.ORA` file, as specified in ["Configuring the Network with the Local Naming Method"](#) on page 6-12.

Note: Keep a copy of the `TNSNAMES.ORA` file, preferably in `$ORACLE_HOME/network/admin` directory. You may need to use this file again later to load net service names into the NIS map.

2. Convert the contents of the `TNSNAMES.ORA` file to a `tnsnames` map using the `tns2nis` program.

Note: The `tns2nis` program is supplied with NIS External Naming on the Oracle Universal Installer tape or disk.

For example, run `tns2nis` on the command line with one argument:

```
tns2nis tnsnames.ora
```

`tns2nis` reads the `NATIVE.ORA` file from the current directory. (If `TNSNAMES.ORA` is not located in the current directory, you can use a full path name to specify its location—for example, `/etc/native.ora` or `$ORACLE_HOME/network/admin/tnsnames.ora`).

The "tnsnames" map is then written into the current working directory.

3. Copy `tnsnames` to the NIS server, if it is not already there.
4. Install the `tnsnames` map using `makedbm`, which is an NIS program. Refer to your NIS documentation for more information.

Note: This step should be performed by the person in charge of NIS administration.

`makedbm` converts the `tnsnames` map into two files that the NIS server can read. The location of these files is platform-specific. Refer to your platform-specific documentation for details.

For example, to generate and install a `tnsnames` map on the Sun Solaris, as root enter the following at the command line

```
# makedbm tnsnames /var/yp/'domainname'/tnsnames
```

Verifying that the "tnsnames" Map Has Been Properly Installed You can test the NIS server to see if the map has been installed properly by typing a command with the format:

```
ypmatch net_service_name tnsnames
```

For example, you might enter:

```
ypmatch payroll.com tnsnames
```

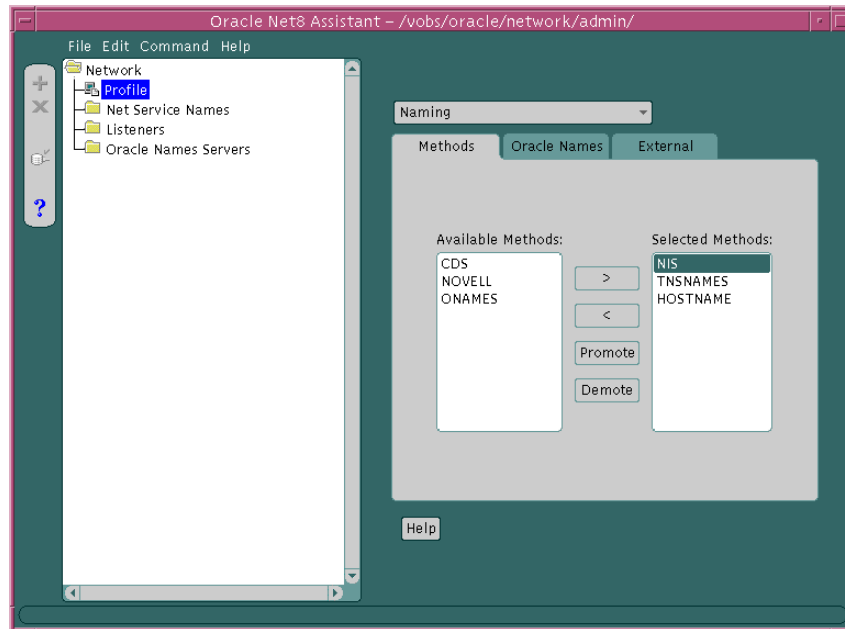
This returns the length of the address (in characters) followed by the address; for example:

```
99 (description=(address=(protocol=tcp)
(host=garlic)(port=1999)))
(connect_data=(service_name=dirprod)))
```

Step 2: Configure Clients

To configure clients, configure NDS as the first method specified in the NAMES_DIRECTORY_PATH parameter in the SQLNET.ORA file. This parameter specifies the order of naming methods Net8 will use to reconcile services.

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Click the Profile icon.
3. Choose Naming from the drop-down list box.
4. Click the Methods tab.
5. Choose NIS from the Available Methods list, then click ">".
6. Select NIS in the Selected Methods list, then use the Promote button to move the selection to the top of the list:



7. Choose Save Network Configuration from the File menu.

8. Choose Exit from the File menu to exit the Net8 Assistant application.

The SQLNET.ORA should contain an entry that looks like the following:

```
names.directory_path=(nis, tnsnames, hostname)
```

Configure NDS

NDS External Naming allows you to use native NDS naming conventions to connect to an Oracle database on a Novell NDS-enabled network. After the NDS External Naming has been installed on clients and servers, users can enter:

```
username/password@net_service_name
```

NDS External Naming provides network users with the following benefits:

- Allows clients to use simple NDS names (partial or full) when connecting to a database.
- Simplifies the maintenance of the TNS addresses; one change will affect all clients using NDS External Naming.
- Reduces network traffic by eliminating the need for the listener to advertise.

Client Operations

The NDS External Naming resides on the client workstation and translates the NDS object name into a TNS address. The client code gets attributes from the NDS tree for the NDS object whose name matches the Oracle net service name. This name can be a full name or a partial name. If it is a partial name, it will be qualified with respect to the *current name context*.

NetWare Server Operations

There are three aspects to the server-side NDS External Naming:

- [Schema Extension](#)
- [SAP \(Service Address Protocol\) Disabling](#)
- [Net Service Name and Address Storage in NDS](#)

Schema Extension During the Oracle installation process on a NetWare 4 server the NDS schema is extended to include an object class called "ORACLE:DBInstance". For NDS External Naming to function, this class will need an attribute called "ORACLE:TNSAddress". If the class does not exist, it is created and will include the TNSAddress attribute. If the class exists but the TNSAddress attribute does not, the

class will be modified. The NLM that performs this during installation is called ORASCHEM.NLM.

SAP (Service Address Protocol) Disabling The SPX protocol for NetWare looks for a value in CONFIG.ORA called ORACLE_SAP. The value of this parameter is ON or OFF. If this parameter is not specified, the default is ON. This has performance implications for SPX networks. When ORACLE_SAP is ON, the SQL*Net listener advertises its address using SAP (Service Advertising Protocol). When ORACLE_SAP is set to OFF, the SQL*Net listener relies on NDS to deliver information to its clients.

If your network consists entirely of NDS enabled clients (that is, clients using NetWare 4 and above), you will get better network performance if you edit the CONFIG.ORA file to set ORACLE_SAP to OFF.

Net Service Name and Address Storage in NDS When the listener is started, it stores its address in the NDS database by locating the Oracle database instance that resides on its server.

Note: An NDS object for the Oracle database must have already been created.

At that point, the address(es) is (are) accessible to the client from the NDS database

System Requirements

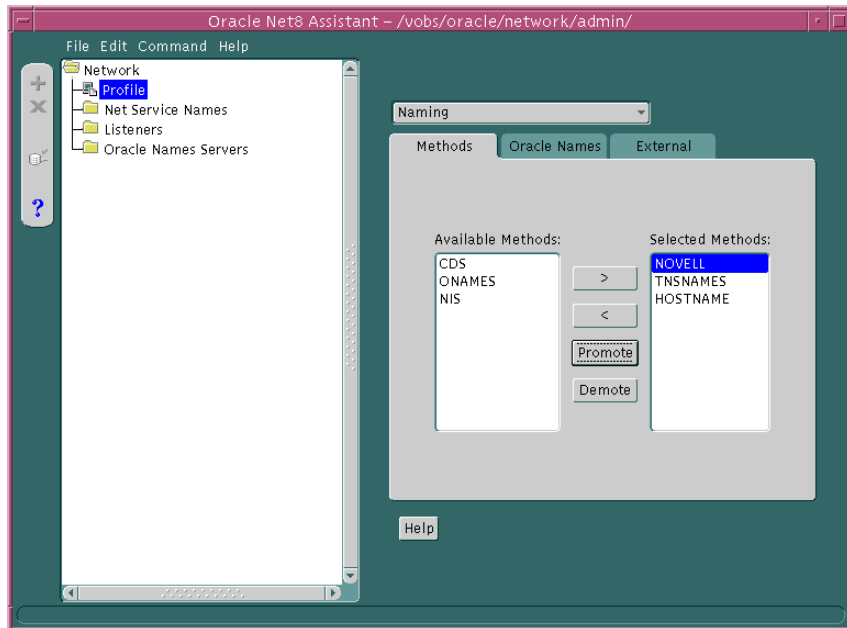
NDS External Naming requires SQL*Net 2.2 or later and Oracle 7.2 or later. It can be used with any client running Novell libraries, but requires NetWare 4.1 or later on the server.

Configure Clients

To configure clients:

1. Configure NDS as the first method specified in the NAMES_DIRECTORY_PATH parameter in the SQLNET.ORA file. This parameter specifies the order of naming methods Net8 will use to reconcile services.
 - a. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
 - b. Click the Profile icon.
 - c. Choose Naming from the drop-down list box.
 - d. Click the Methods tab.
 - e. Choose NDS from the Available Methods list, then click ">".

- f. Select Novell in the Selected Methods list, then use the Promote button to move the selection to the top of the list:



- g. Choose Save Network Configuration from the File menu.
- h. Choose Exit from the File menu to exit the Net8 Assistant application.

The SQLNET.ORA file should contain an entry that lists NOVELL first in the NAMES.DIRECTORY_PATH parameter:

```
names.directory_path=(novell, tnsnames, hostname)
```

2. Optionally, configure a typeless or typed name.

- To specify a typeless name:

Add the following parameter to the SQLNET.ORA file to specify the default name context in which to look for the name to be resolved.

```
native_names.nds.name_context
```

Note: You must add this parameter manually to the SQLNET.ORA file. It cannot be created using Net8 Assistant.

For example, if the name of the database object is "Payroll.Finance.Oracle" and the SQLNET.ORA parameter is `NATIVE_NAMES.NDS.NAME_CONTEXT=Finance.Oracle`, then the name "Payroll" will be qualified to ".Payroll.Finance.Oracle". This is an example of a typeless name.

Note: The leading dot designates this as a full NDS name. If you want to override the name context parameter in SQLNET.ORA, then you can specify the full NDS name in the connect string by using a leading dot.

- To specify a typed name, enter a parameter and value in SQLNET.ORA like the following:

```
native_names.nds.name_context=OU=Finance.O=Oracle
```

This line will be parsed to produce the typed name

```
CN=Payroll.OU=Finance.O=Oracle.
```

This parameter works similarly to the NET.CFG parameter "name context". The name context in SQLNET.ORA will override the entry in NET.CFG. If the SQLNET.ORA parameter is not specified, the NET.CFG parameter will be used. If no name context is specified in either file, it defaults to [root]. See the Novell client documentation for more information about the NET.CFG parameters.

Note: The default name context specified in SQLNET.ORA cannot contain a leading dot. This will result in an NDS error code of -309 (ERR_EXPECTED_IDENTIFIER):

The parameter being parsed is not typed.

Configure the Server

To configure the NetWare Server:

1. Configure NDS as the first method specified in the NAMES_DIRECTORY_PATH parameter in the SQLNET.ORA file. This parameter specifies the order of naming methods Net8 will use to reconcile services.
 - a. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
 - b. Click the Profile icon.
 - c. Choose Naming from the drop-down list box.
 - d. Click the Methods tab.
 - e. Choose Novell from the Available Methods list, then click ">".
 - f. Select Novell in the Selected Methods list, then use the Promote button to move the selection to the top of the list.
 - g. Choose Save Network Configuration from the File menu.
 - h. Choose Exit from the File menu to exit the Net8 Assistant application.
2. Optionally, add the following parameter to the CONFIG.ORA file located in `$ORACLE_HOME/nlm`:

```
oracle_sap=[off|on]
```

ORACLE_SAP can be set to either ON or OFF. When ORACLE_SAP is ON, the listener advertises its address using SAP (Service Advertising Protocol). When ORACLE_SAP is set to OFF, the listener relies on NDS to deliver information to its clients. If not specified in CONFIG.ORA, ORACLE_SAP defaults to on. To reduce network traffic on a network where all clients use NDS, use:

```
oracle_sap=off
```

Known Limitations

Following are some known limitations when using the NDS External Naming:

- The TNS address stored in the NDS database cannot be more than 2048 characters in length.
- You cannot use more than one listener per database instance. If you do, the last listener to start will overwrite any other TNS address stored in the database object.
- If SID support is enabled on the server, you should not use a NULL SID for any of the database instances. If a NULL SID is used for one of the instances, you cannot connect to it using SQL*Net version 2 or Net8.
- If SID support is not enabled, the last SID specified in the listener's SID_LIST will be the one used. In this case, the SID is transparent to the user and database. The user does not see it and the database ignores it.

Note: SID support is controlled by the following parameter in the CONFIG.ORA file:

```
nw_enable_sid_support=[true|false]
```

Configuring the Listener

Services coordinate their session with the help of a listener. The listener receives connection requests on behalf of a client application. Listeners are configured to "listen on" an address specified in a listener configuration file, LISTENER.ORA, for a database service or non-database service. Once the LISTENER.ORA file is configured, the listener may be managed with the Listener Control Utility (LSNRCTL).

An Oracle8i database instance registers itself with the listener. Prior releases required configuration of the database SID description information. In order to take advantage of release 8.1 features, use the latest listener.

By default the following listening endpoints are configured automatically during installation:

- IPC for connections to external procedures
- TCP/IP for a Net8 client connection to the Oracle database
- TCP/IP for a IIOP client connection to the Java option

Additionally, the SID_LIST is automatically configured for the database and external procedures.

The default LISTENER.ORA is shown below with annotations of content:

```
listener=
  (description_list=
    # ipc listenening endpoint for external procedures' connections
    (description=
      (address_list=
        (address=(protocol=ipc)(key=extproc0))
      )
    )
    # tcp/ip listenening endpoint for java option connections
    (description=
      (protocol_stack=
        (presentation=giop)
        (session=raw)
      )
      (address_list=
        (address=(protocol=tcp)(host=sales-pc)(port=2481))
      )
    )
  )
```



```

# tcp/ip listenening endpoint for database connections
(description=
  (address_list=
    (address=(protocol=tcp)(host=sales-pc)(port=1521))
  )
)
)
# list of services for a listener named LISTENER to receive requests for
sid_list_listener=
(sid_list=
  # service description for external procedures
  (sid_desc=
    (sid_name=plsextproc)
    (oracle_home=c:\orahome81)
    (program=extproc)
  )
  # service description for the database specified
  (sid_desc=
    (global_dbname=sales.com)
    (oracle_home=c:\orahome81)
    (sid_name=sales)
  )
)
)

```

The following listener configuration items are addressed in this section:

- [Re-Starting the Listener After a LISTENER.ORA File Modification](#)
- [Configuring Listener Protocol Addresses](#)
- [Using a Listener Name Other than LISTENER](#)
- [Handling Larger Volumes of Connection Requests](#)
- [Configuring a Listener with Service Information](#)

See also the following:

- ["Configuring Net8 for External Procedures"](#) on page 7-65 for further information about external procedure configuration
- ["Configuring Java Option Connections"](#) on page 7-50 for further information about Java option and IIOP client configuration

Re-Starting the Listener After a LISTENER.ORA File Modification

If you modify the LISTENER.ORA, the listener must be stopped and re-started with the LSNRCTL utility.

To stop the listener:

```
lsnrctl  
lsnrctl> start [listener_name]
```

To stop the listener:

```
lsnrctl  
lsnrctl> stop [listener_name]
```

listener_name is the name of the listener defined in the LISTENER.ORA file. It is not necessary to identify the listener if you are using the default listener, named LISTENER.

For further information, see:

- ["Net8 Component Startup Overview"](#) on page 8-2
- ["Listener Control Utility \(LSNRCTL\)"](#) on page A-3

Configuring Listener Protocol Addresses

The listener address you configured for either local naming or Oracle Names must also be configured in LISTENER.ORA. During a Typical or Minimal installation, TCP/IP is automatically configured in the LISTENER.ORA. The other protocols, however, are not configured. To configure other protocol listener addresses:

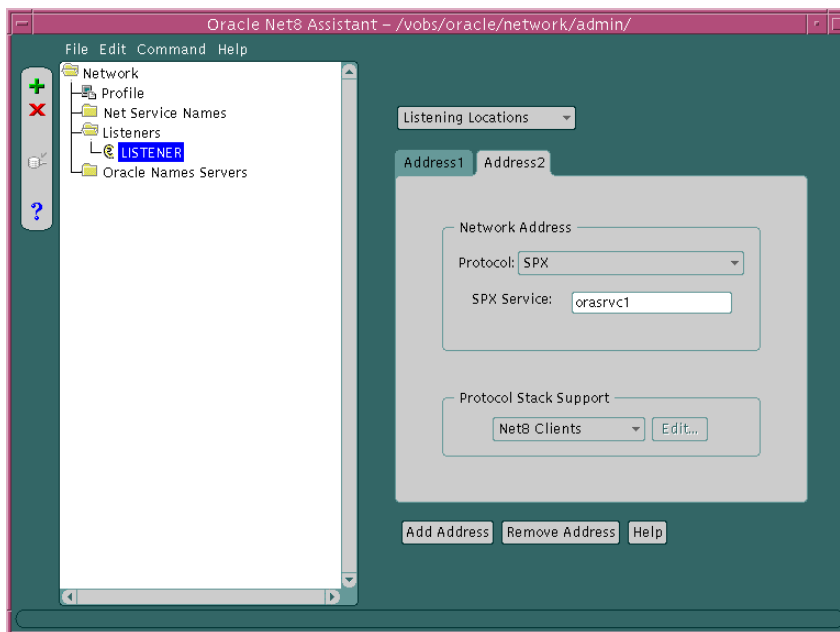
1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Double-click on the Listeners folder.
3. Click on a listener.

Note: If a LISTENER.ORA has been configured during installation and listener has never been started with the LSNRCTL control utility, the Listeners folder will not display a listener. If this occurs, exit the Net8 Assistant, start the listener (as described in "[Net8 Component Startup Overview](#)" on page 8-2), and start the Net8 Assistant again.

If a listener does not exist because a LISTENER.ORA was not installed:

- a. Select Create from the Edit menu or click on the "+" button to create a new LISTENER.ORA file.
 - b. Enter a unique listener name in the *Choose Listener Name* dialog box.
4. Choose Listening Locations from the drop-down list box.
 5. Click Add Address.

- Choose the protocol and enter the requested protocol address information, as described on ["Configuring Protocol Addresses"](#) on page 6-76:



- Choose Net8 Client for typical client connections to database services, or choose IIOP Clients for client connection to the Java option.

When configuring the listener to listen on TCP/IP for typical Net8 clients, you should choose the default port or 1521. If you do not, you must configure the LOCAL_LISTENER parameter in the INITSID.ORA file and resolve the listener name through a naming method. For further information about configuring non-default port numbers in LISTENER.ORA for Net8 clients, see ["Configuring a Listener that Uses a Non-Default Port"](#) on page 6-67.

If IIOP Clients was chosen, ensure the protocol was TCP/IP on port 2481 or TCP/IP with SSL on port 2482. If you do not, you must configure the LOCAL_LISTENER parameter or the MTS_DISPATCHERS parameter's LISTENER attribute in the INITSID.ORA file, and resolve the listener name through a naming method. For further information about configuring non-default port numbers in LISTENER.ORA for IIOP clients, follow the steps in ["Configuring a Non-Default Listener"](#) on page 7-59.

8. Repeat Step 5-7 for additional protocols.
9. Select Save Network Configuration from the File menu.
10. Choose Exit from the File menu to exit the Net8 Assistant application.

For example, a LISTENER.ORA file statically configured with the for SPX address for service named oraservc1 in addition to the default TCP/IP address is shown below:

```
listener=
  (address=
    (protocol= tcp)
    (host= sales-pc)
    (port= 1521)
  )
  (address=
    (protocol= spx)
    (service=orasrvcl)
  )
```

Configuring a Listener that Uses a Non-Default Port

If you configure a TCP/IP listener addresses that does not use the default 1521 port for Net8 clients, you must configure the LOCAL_LISTENER parameter in the INITSID.ORA file and resolve it with a net service name entry in the TNSNAMES.ORA file or Oracle Names server.

Configuring the LOCAL_LISTENER Parameter

The LOCAL_LISTENER parameter should be configured as follows:

```
local_listener=listener_name
```

In order for the dispatcher to register its information with a non-default listener, the listener name you specified for the LOCAL_LISTENER parameter must be resolved through a naming method, such as a TNSNAMES.ORA file on the server or an Oracle Names server.

The entry should only contain the address of the listener, not the service name information in the CONNECT_DATA portion.

Configuring a TNSNAMES.ORA File

The Net8 Assistant does not allow you to configure a TNSNAMES.ORA without the CONNECT_DATA information. Therefore, Oracle Corporation recommends you create the TNSNAMES.ORA manually.

If the listener name is LISTENER1, the listening endpoint uses TCP/IP on port 1421, and the chosen naming method is the local naming method, the entry in the TNSNAMES.ORA would be:

```
listener1.us.oracle.com=
(description=
  (address=
    (protocol= tcp)
    (host= sales-pc)
    (port= 1421)
  )
)
```

Multiple addresses are supported, but multiple address connect-time failover and client load balancing features are not supported. Oracle Connection Manager's SOURCE_ROUTE parameter is supported.

Configuring an Oracle Names Server

An entry for the listener can be made through an Oracle Names server through the Net8 Assistant:

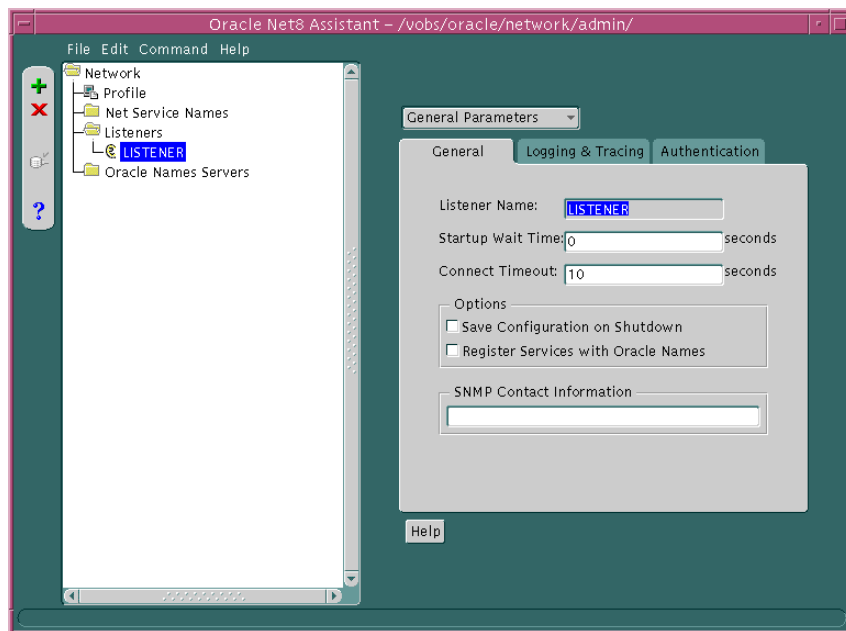
1. Start Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Select the Oracle Names Servers folder.
3. Click on an Oracle Names server.
4. Select Manage Data from the drop-down list box.
5. Select the Advanced tab:
6. Click Add, and enter the listener name in the Name field, A.SMD record type in the Type field, and enter the address in the Value field, such as:
`(address=(protocol= tcp)(host= sales-pc)(port= 2481))`
7. Choose Save Network Configuration from the File menu.
8. Choose Exit from the File menu to exit the Net8 Assistant.

Using a Listener Name Other than LISTENER

By default the listener name is LISTENER. The listener name must be unique on any given node. The listener can be given any name.

To provide a listener with a new name:

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Double-click on the Listeners folder.
3. Click on a listener.
4. Choose General Parameters from the drop-down list box:



5. Modify the Listener Name field with the new listener name.
6. Select Save Network Configuration from the File menu.
7. Choose Exit from the File menu to exit the Net8 Assistant application.

Handling Larger Volumes of Connection Requests

If you expect the listener to handle large volumes of connection requests, you may specify a queue for the process. This will allow the listener to dynamically handle larger numbers of concurrent connection requests.

To specify a queue size for a listener, enter a value to the `QUEUESIZE` parameter at the end of any listening address:

```
listener=  
  (address=  
    (protocol= tcp)  
    (host= sales-pc)  
    (port= 1521)  
    (queuesize=20)  
  )
```

Note: Currently, you can only configure the queue size for listeners operating on TCP/IP and DECnet. The default queue size is system specific. On Solaris, the default queue size is set to 5. The queue size is 5 for Windows NT 4.0 Workstation and 50 for Windows NT 4.0 Server.

Configuring a Listener with Service Information

In order for the listener to accept client requests to an Oracle8i release 8.0 or Oracle7 database, the LISTENER.ORA file must be configured with information about the database. Static configuration is also required for other services, such as external procedures and heterogeneous services, and some management tools, including Oracle Enterprise Manager.

See [Chapter 7, "Enabling Optional Net8 Features"](#) for more information about statically configuring the listener for external procedures and heterogeneous services. See the *Oracle Enterprise Manager Configuration Guide* for Oracle Enterprise Manager configuration.

CAUTION: Implementing connect-time failover does not allow use of static service configuration parameters in the LISTENER.ORA file. However, static configuration is required for Oracle8i release 8.0 or Oracle7 databases and Oracle Enterprise Manager. See ["Connect-Time Failover"](#) on page 2-29 for further information about connect-time failover and the *Oracle8i Parallel Server Setup and Configuration Guide* for additional configuration instructions on setting up connect-time failover for Oracle Parallel Server in an Oracle Enterprise Manager environment.

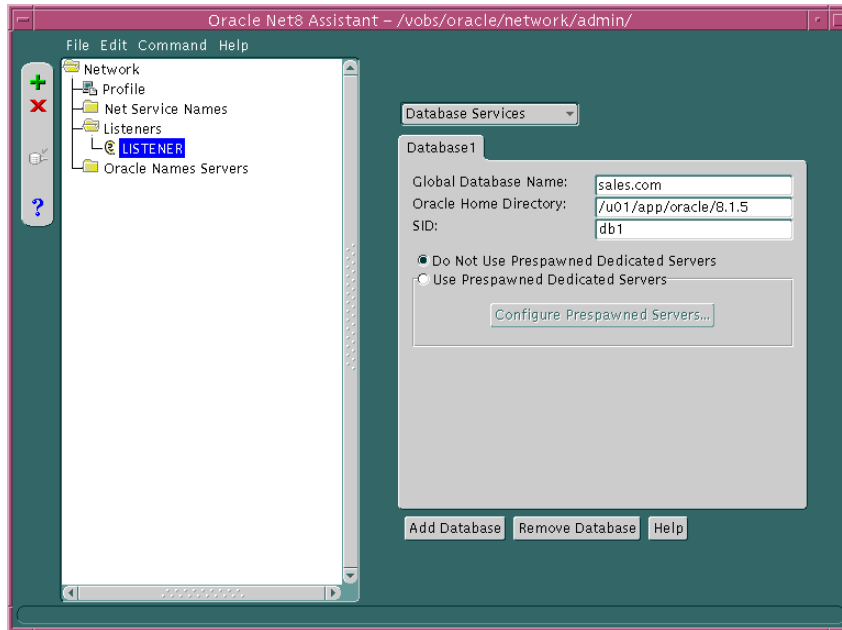
To configure information about database instances that the listener is servicing, you will need to provide the following information in the listener configuration file:

Net8 Assistant Option	LISTENER.ORA Parameter	Description
System Identifier (SID)	SID_NAME	The system identifier or SID is the Oracle system ID for the database server, as provided by the INSTANCE_NAME parameter in the database initialization parameter file (INITSID.ORA). Note: This parameter value should match that of the INSTANCE_NAME parameter.
Global Database Name	GLOBAL_DBNAME	The global database name is the name and domain name of the database as provided by the SERVICE_NAMES parameter in the INITSID.ORA file. If you want to refer to the database by its global database name on the network, then you must specify that global database name to the listener. Note: This parameter value should match the value of the SERVICE_NAMES parameter.

Net8 Assistant Option	LISTENER.ORA Parameter	Description
Oracle Home Directory	ORACLE_HOME	The Oracle Home Directory identifies the Oracle Home location of the database that you are specifying. Note: Not required for Windows NT.

To statically configure the listener:

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Double-click on the Listeners folder.
3. Click on a listener.
4. Choose Database Services from the drop-down list box:



5. Modify the fields as appropriate.
6. Select Save Network Configuration from the File menu.
7. Choose Exit from the File menu to exit the Net8 Assistant application.

An excerpt of a LISTENER.ORA file statically configured with the `SID_LIST_listener_name` section is shown below:

```
sid_list_listener=(sid_list=
(sid_desc=
(global_dbname=sales.com)
(sid_name=db1)
(oracle_home=/u01/app/oracle/8.1.5)
)
```

Configuring Prestarted or Prespawnd Dedicated Servers

Note: Prespawnd dedicatess servers are not available for Windows NT.

To create prespawnd dedicated servers, add the following four parameters in each `SID_DESC` in the LISTENER.ORA file:

Net8 Assistant Option	LISTENER.ORA Parameter	Description
Maximum Prespawnd Servers	PRESPAWN_MAX	The maximum number of prespawnd dedicated servers the listener will create. This number must be at least as many as the sum of the pool size for each protocol. Set this value to a large number so that prespawnd dedicated servers are always available for new connections.
TCP/IP, SPX, Named Pipes, IPC	PROTOCOL	The protocol on which the listener creates prespawnd dedicated servers.
Number	POOL_SIZE	The number of unused prespawnd dedicated servers for the listener to maintain on the selected protocol. Choose a number that is greater than 0 but no greater than the PRESPAWN_MAX value. The value should be about what you expect the average number of connections to be at any given time.

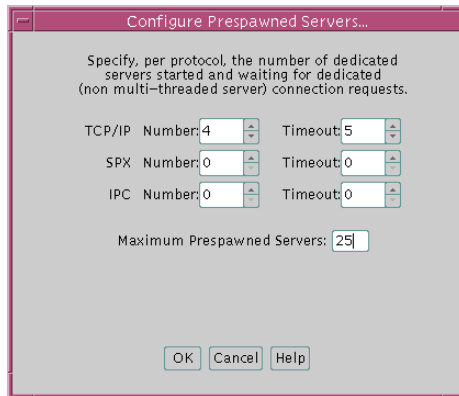
Net8 Assistant Option	LISTENER.ORA Parameter	Description
Timeout	TIMEOUT	Time in minutes that an inactive prespawnded dedicated server process waits for the next connection. The value should be greater than 0. (A value of 0 will allow an inactive shadow process to continue indefinitely, thus wasting machine resources.) Set a short time out value. The time out is activated only after a prespawnded dedicated server process has carried a connection and been disconnected. In other words, prespawnded dedicated servers that are waiting for their first connection do not time out.

Note: These parameters are not supported on Windows NT.

To configure prespawnded dedicated servers:

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Double-click on the Listeners folder.
3. Click on a listener.
4. Choose Database Services from the drop-down list box.
5. Select Use Prespawnded Dedicated Servers, and click Configure Prespawnded Servers.

6. The *Configure Prespawnd Servers* dialog box appears:



7. In the *Configure Prespawnd Servers* dialog box, enter appropriate information.

8. Select Save Network Configuration from the File menu.

9. Choose Exit from the File menu to exit the Net8 Assistant application.

An excerpt of a LISTENER.ORA file configured with prespawnd dedicated server is shown below:

```

sid_list_listener=(sid_list=
(sid_desc=
(global_dbname=sales.com)
(sid_name=db1)
(oracle_home=/u01/app/oracle/product/8.1.5)
(prespawn_max=25)
(prespawn_list=
(prespawn_desc=
(protocol=tcp)
(pool_size=4)
(timeout=5)
)
)
)
)
)

```

Configuring Protocol Addresses

When a connection is made, the client and the receiver of the request, such as a listener, Oracle Names server, or Oracle Connection Manager, are configured with identical protocol addresses. The purpose of the address is to identify the network address of a network object. The client uses this address to send the connection request to a particular network object location, and the recipient "listens" for requests on this address, and grants a connection based on its address information matching the client information. It is important to install the same protocols for the client and the connection recipient, as well as configure the same address.

This section covers the following topics:

- [Address Components](#)
- [Protocol Parameters](#)

Address Components

Protocol addresses are contained within the following components:

ADDRESS

Purpose: Defines a protocol address. This parameter may be embedded under ADDRESS_LIST or DESCRIPTION. A DESCRIPTION is typically only used in a TNSNAMES.ORA or LISTENER.ORA file.

Additional Information: Each protocol has its own required parameters needed in the address, as explained in "[Protocol Parameters](#)" on page 6-78.

Example:

```
(address=
(protocol=tcp)
(host=sales-pc)
(port=1521)
)
```

ADDRESS_LIST

Purpose: Defines lists of addresses that share a set of common characteristics.

This parameter may be embedded under `DESCRIPTION` or `DESCRIPTION_LIST`. A `DESCRIPTION` or `DESCRIPTION_LIST` is typically only used in a `TNSNAMES.ORA` or `LISTENER.ORA` file.

For a description of `DESCRIPTION` and `DESCRIPTION_LIST`, see "[DESCRIPTIONS and DESCRIPTION_LISTs](#)" on page C-38.

Note: If there is only one list in a `TNSNAMES.ORA` or a `LISTENER.ORA` file, `ADDRESS_LIST` is not necessary.

Example:

```
(address_list=  
  (address=  
    (protocol=tcp)  
    (host=sales-pc)  
    (port=1521)  
  )  
  (address=  
    (protocol=tcp)  
    (host=hr-pc)  
    (port=1521)  
  )  
)  
(address_list=  
  (address=  
    (protocol=tcp)  
    (host=finance-pc)  
    (port=1521)  
  )  
  (address=  
    (protocol=tcp)  
    (host=sales-pc)  
    (port=1521)  
  )  
)  
)
```

Protocol Parameters

The listener, Oracle Names server and Oracle Connection Manager are identified by protocol addresses. The table below describes the parameters used by the supported Oracle protocols:

Protocol	Parameter	Description
All	PROTOCOL	Indicates the type of network on which the TNS-based application resides.
Bequeath ¹	PROGRAM	Identifies the Oracle8i executable.
Bequeath	ARGV0	Identifies the service name.
Bequeath	ARGS	Identifies the source of the connection (local client).
IPC	KEY	Indicates a way of identifying the server. Oracle Corporation recommends using the name of the service.
Named Pipes	SERVER	Indicates the name of your Oracle8i server computer.
Named Pipes	PIPE	Indicates the pipe name you use to connect to your server (the same PIPE keyword you specified on server with Named Pipes). This name can be any arbitrary name.
LU6.2	LU_NAME	Identifies the Oracle8i server; must be a fully-qualified name.
LU6.2	LLU or LOCAL_LU	Identifies the local LU alias. This parameter cannot be used with LLU_NAME.
LU6.2	LLU_NAME or LOCAL_LU_NAME	Specifies the local LU name; must be a fully qualified name. This parameter cannot be used with LLU.
LU6.2	MODE or MDN	Identifies the log mode entry of the LU6.2 network; the value is typically ORAPLU62.
LU6.2	PLU or PARTNER_LU_NAME	Identifies the Oracle8i server; must be a fully qualified name. This parameter cannot be used with PLU_LA.
LU6.2	PLU_LA or PARTNER_LU_LOCAL_ALIAS	Identifies the partner LU alias of the Oracle7 Server. This parameter cannot be used with PLU.

Protocol	Parameter	Description
LU6.2	TP_NAME or TPN	Identifies the transaction program name of the host machine. This parameter is not required for a connection to an MVS host.
SPX	SERVICE	Defines the name of the TNS-based application on the network. (Mandatory for server and client.) Speak to your network administrator to learn the service names of TNS-based applications on your network.
TCP/IP and TCP/IP with SSL	HOST	Identifies the server.

Protocol	Parameter	Description
TCP/IP and TCP/IP with SSL	PORT	<p>Identifies the listening port number:</p> <ul style="list-style-type: none"> ■ 1521 is the default listening port for client connections to a database service. In future release, this port number may change to the officially registered port number of 2483 for TCP/IP and 2484 for TCP/IP with SSL. ■ 2481 is the recommended and officially registered listening port for client connections to the Java option using TCP/IP. ■ 2482 is the recommended and officially registered listening port for client connections to the Java option using TCP/IP with SSL. ■ 1575 is the default and officially registered listening port for an Oracle Names server using TCP/IP or TCP/IP with SSL. ■ 1630 is the default and officially registered listening port using TCP/IP for the Oracle Connection Manager CMGW gateway process. ■ 1830 is the default and officially registered listening port using TCP/IP for the Oracle Connection Manager CMADMIN administrative process. <p>Note: Oracle Corporation allows port numbers from 1 to 65535, but port numbers less than 1024 are restricted to use by privileged processes on many operating systems.</p>

¹ Bequeath requires configuration information in TNSNAMES.ORA but not in LISTENER.ORA.

Enabling Optional Net8 Features

This chapter describes optional features you can enable.

Specific topics discussed are:

- [Modifying Net Service Names](#)
- [Configuring a Profile](#)
- [Enabling Connection Concentration](#)
- [Enabling Multi-Protocol Support](#)
- [Enabling Net8 Access Control](#)
- [Configuring the Client for Oracle Connection Manager Features](#)
- [Enabling Connection Pooling](#)
- [Configuring Java Option Connections](#)
- [Configuring Non-Oracle Database Services](#)

Modifying Net Service Names

This sections describes the following:

- [Modifying Attributes of a Net Service Name](#)
- [Renaming a Net Service Name](#)
- [Adding a Protocol Address to a Net Service Name](#)
- [Deleting a Protocol Address from a Net Service Name](#)
- [Deleting a Net Service Name](#)
- [Configuring Advanced Connection Attributes for a Net Service Name](#)
- [Configuring Multiple Address Options](#)

These changes are saved to the TNSNAMES.ORA file at `$ORACLE_HOME/network/admin` on UNIX or `ORACLE_HOME\network\admin` on Windows platforms.

Modifying Attributes of a Net Service Name

To modify attributes of a net service name, such as its name, service name, SID, or protocol address:

1. Start the Net8 Configuration Assistant:
 - On UNIX, run `netca` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Configuration Assistant.
2. Select *Local Net8 Service Name configuration*, then click Next.
3. Click Modify.
4. Follow the prompts in the wizard and online help to modify any of the attributes.
5. Click Finish to save your configuration and dismiss the Net8 Configuration Assistant application.

Renaming a Net Service Name

To change the name of a net service name:

1. Start the Net8 Configuration Assistant:
 - On UNIX, run `netca` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Configuration Assistant.
2. Select *Local Net8 Service Name configuration*, then click Next.
3. Click Rename.
4. Follow the prompts in the wizard and online help to modify any of the attributes.
5. Click Finish to save your configuration and dismiss the Net8 Configuration Assistant application.

Adding a Protocol Address to a Net Service Name

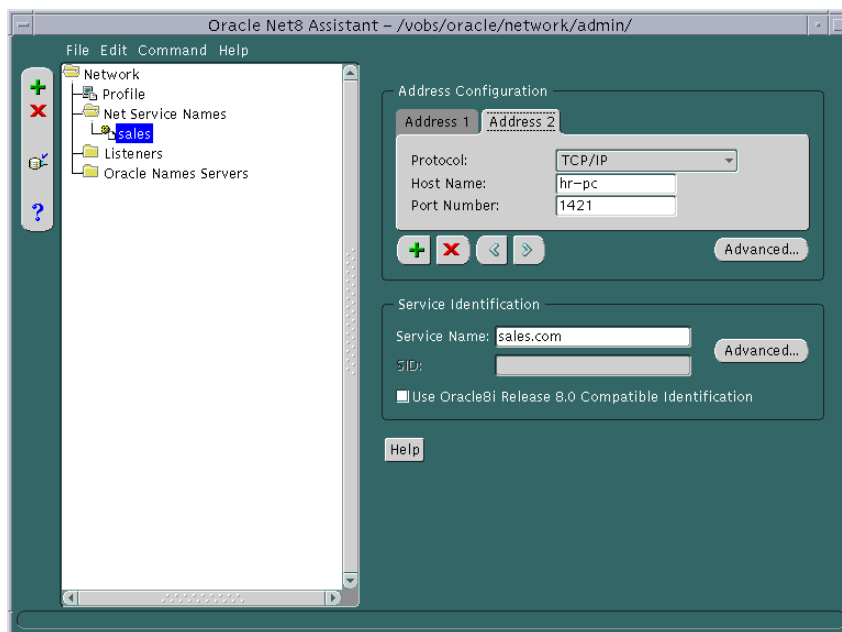
Each connect descriptor can contain one or more listener addresses:

```
net_service_name=
(description=
  (address= (protocol_address_information))
  (address= (protocol_address_information))
  (address= (protocol_address_information))
  (connect_data=
    (service_name=service_name)
  )
)
```

To add an additional protocol address to a net service name:

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Double-click on the Net Service Names folder in the directory tree.
3. Select the net service name.
4. Click "+" in the Address Configuration group box.

A new address tab appears:

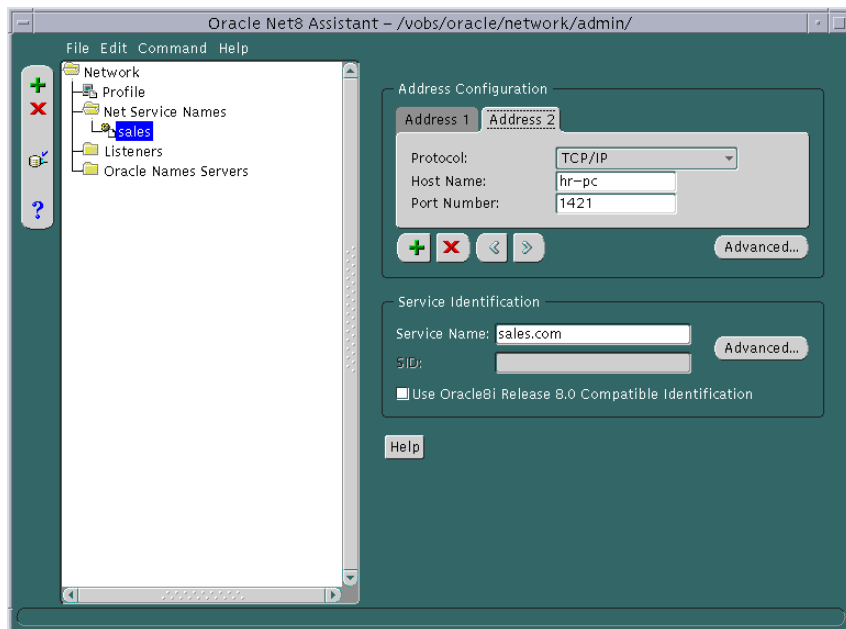


5. Choose the protocol and enter appropriate address information. For more information about protocol address parameters, see "[Configuring Protocol Addresses](#)" on page 6-76.
6. Order the address according to where it should be in the address list with the "<" and ">" buttons. Unless multiple address options are configured, the first address in the list is contacted first. See "[Configuring Multiple Address Options](#)" on page 7-11 to configure multiple address options.
7. Choose Save Network Configuration from the File menu.
The TNSNAMES.ORA file is re-created.
8. Choose Exit from the File menu.
The Net8 Assistant application exits.

Deleting a Protocol Address from a Net Service Name

To delete a protocol address from a net service name in the TNSNAMES.ORA file:

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Double-click the Net Service Names folder in the directory tree.
3. Select the net service name.
4. Click "X" in the Address Configuration group box:



5. Choose Save Network Configuration from the File menu.
The TNSNAMES.ORA file is re-created.
6. Choose Exit from the File menu.
The Net8 Assistant application exits.

Deleting a Net Service Name

To delete a net service name from the TNSNAMES.ORA file:

1. Start the Net8 Configuration Assistant:
 - On UNIX, run `netca` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Configuration Assistant.
2. Select *Local Net8 Service Name configuration*, then click Next.
3. Click Delete.
4. Follow the prompts in the wizard and online help to delete the net service name.
5. Click Finish to save your configuration and dismiss the Net8 Configuration Assistant application.

If you click Finish, the net service name is deleted, and the TNSNAMES.ORA file is re-created.

Configuring Advanced Connection Attributes for a Net Service Name

A TNSNAMES.ORA's CONNECT_DATA section of a connect descriptor may optionally contain the following features:

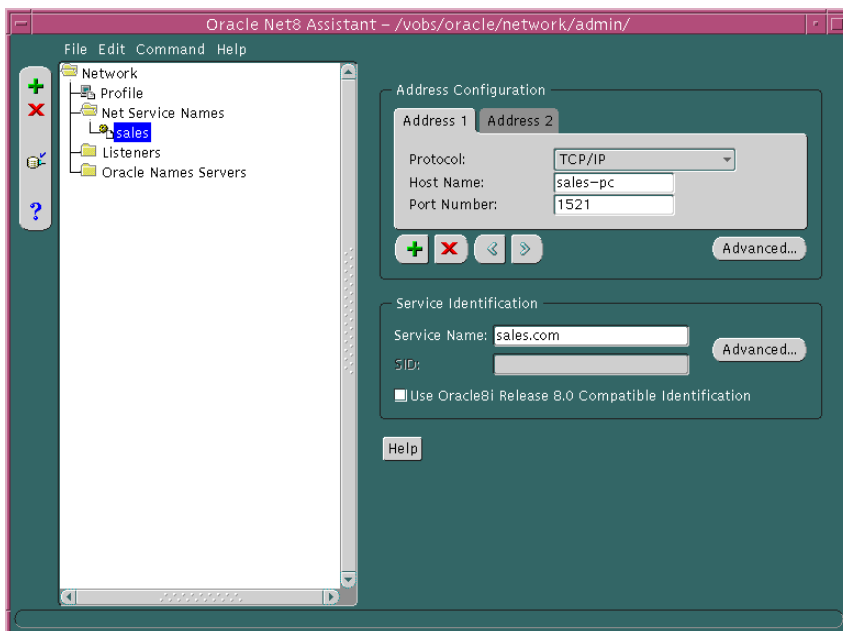
Net8 Assistant option	TNSNAMES.ORA Parameter	Description
Instance Name	INSTANCE_NAME	<p>Identifies the database instance to access.</p> <p>The instance name can be obtained from the INSTANCE_NAME parameter in the INITSID.ORA file.</p> <p>Note: Not enabled if <i>Use Oracle8i Release 8.0 Compatible Identification</i> is turned ON.</p> <p>Additional Information: See "Setting Service Names and Instance Names" on page 6-3.</p>
Handler Name	HANDLER_NAME	<p>Identifies a service handler, such as an MTS dispatcher, to access.</p> <p>Note: Not enabled if <i>Use Oracle8i Release 8.0 Compatible Identification</i> is turned ON.</p> <p>Note: Not enabled for release 8.1.5.</p>
Session Data Unit (SDU) Size	SDU	<p>If you want to optimize the transfer rate of data packets being sent across the network, you can specify the session data unit (SDU) size to change the performance characteristics having to do with the packets sent across the network.</p> <p>Additional Information: See "Optimizing Data Transfer by Adjusting the Session Data Unit (SDU) Size" on page 3-13.</p>
Use a Dedicated Server	SERVER	<p>If you wish for the client connection to bypass multi-threaded server and to use dedicated server, turn this option on.</p>
Use for Heterogeneous Services	HS	<p>If you wish for an Oracle8i server to access a non-Oracle system, turn this option on.</p> <p>Note: Not enabled if <i>Use Oracle8i Release 8.0 Compatible Identification</i> is turned ON.</p> <p>Additional Information: See "Configuring Net8 for Oracle Heterogeneous Services" on page 7-67.</p>

Net8 Assistant option	TNSNAMES.ORA Parameter	Description
Oracle Rdb Settings		
Oracle RDB Database	RDB_DATABASE	Specifies the file name of an Oracle RDB database. Additional Information: See " Configuring Net8 for an Oracle Rdb Database " on page 7-71.
Type of Service	TYPE_OF_SERVICE	Specifies the type of service to use for an Oracle RDB database. Additional Information: See " Configuring Net8 for an Oracle Rdb Database " on page 7-71.
Global Database Name	GLOBAL_NAME	Identifies the Oracle Rdb database. Additional Information: See " Configuring Net8 for an Oracle Rdb Database " on page 7-71.

To configure these advanced options:

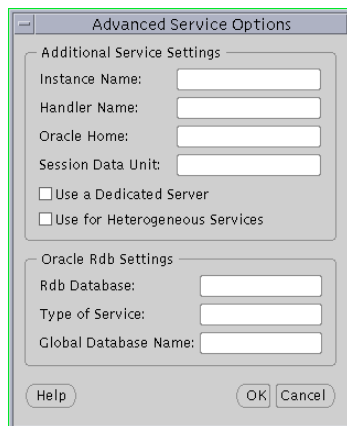
1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Double-click the Net Service Names folder in the directory tree.

3. Select the net service name:



4. Click Advanced in the Service Identification group box.

The *Advanced Service Options* dialog box appears:



5. Enter fields or select options as appropriate, then click OK.
6. Choose Save Network Configuration from the File menu.
The TNSNAMES.ORA file is re-created.
7. Choose Exit from the File menu.
The Net8 Assistant application exits.

Configuring Multiple Address Options

When multiple addresses have been configured for a net service name, the following features may be configured:

- Source Route (SOURCE_ROUTE)
- Client Load Balance (LOAD_BALANCE)
- Connect-Time Failover (FAILOVER)

Option	Description
Source Route (SOURCE_ROUTE)	<p>Instructs Net8 to use each address in the order presented until the destination reached when set to ON. This parameter is required when reaching the destination requires more than one address stop. This feature is typically used to enable Oracle Connection Manager features.</p> <p>Additional Information: For information about using the SOURCE_ROUTE parameter, see:</p> <ul style="list-style-type: none"> ■ "Enabling Connection Concentration" on page 7-31 ■ "Enabling Multi-Protocol Support" on page 7-35 ■ "Enabling Net8 Access Control" on page 7-38
Connect-Time Failover (FAILOVER)	<p>At connect time instructs Net8 to fail over to a different listener if the first listener fails when set to ON. The number of addresses in the list determines how many addresses are tried. When set to OFF, instructs Net8 to try one address.</p> <p>Connect-time failover is turned ON by default for multiple address lists (ADDRESS_LIST), connect descriptors (DESCRIPTION), and multiple connect descriptors (DESCRIPTION_LIST).</p>
Client Load Balance (LOAD_BALANCE)	<p>When set to ON, instructs Net8 to progress through the list of listener addresses in a random sequence, balancing the load on the various listeners. When set to OFF, instructs Net8 to try the addresses sequentially until one succeeds.</p> <p>Client load balancing is turned ON by default for multiple connect descriptors (DESCRIPTION_LIST).</p>

You can achieve the following effects from client load balancing and connect-time failover:

Net8 Assistant Option	TNSNAMES.ORA Parameter
<i>Try each address, in order, until one succeeds</i>	FAILOVER=ON for release 8.1 clients SOURCE_ROUTE=OFF for pre-release 8.0 clients.
<i>Try each address, randomly, until one succeeds</i>	LOAD_BALANCE=ON
Note: Not enabled if <i>Use Options Compatible with Net8 8.0 Clients</i> is turned ON.	FAILOVER=ON
<i>Try one address, selected at random</i>	LOAD_BALANCE=ON
Note: Not enabled if <i>Use Options Compatible with Net8 8.0 Clients</i> is turned ON.	
<i>Use each address in order until destination reached</i>	SOURCE_ROUTE=ON
<i>Use only the first address</i>	LOAD_BALANCE=OFF
Note: Not enabled if <i>Use Options Compatible with Net8 8.0 Clients</i> is turned ON.	FAILOVER=OFF SOURCE_ROUTE=OFF

CAUTION: Implementing connect-time failover does not allow use of static service configuration parameters in the LISTENER.ORA file. However, static configuration is required for Oracle8i release 8.0 or Oracle7 databases and Oracle Enterprise Manager. For further information about statically configuring the listener, see ["Configuring a Listener with Service Information"](#) on page 6-71.

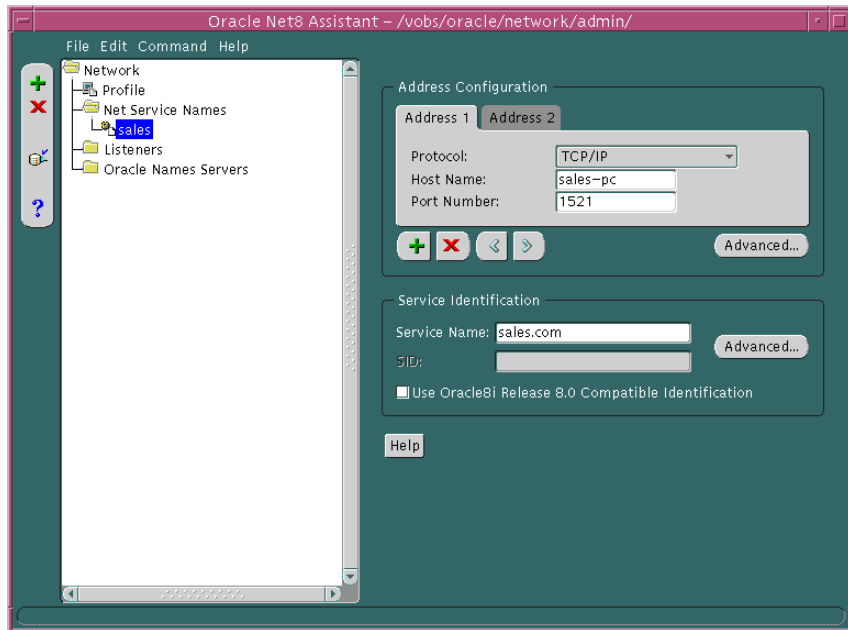
See the *Oracle8i Parallel Server Setup and Configuration Guide* for instructions on setting up connect-time failover for Oracle Parallel Server in an Oracle Enterprise Manager environment.

Note: Connect-time failover and client load balancing options and parameters are only available for release 8.1 configurations. Source routing (*Use each address in order until destination reached*) is available for both pre-release 8.1 and release 8.1 configurations.

Note: It is not possible to set client load balancing or connect-time failover with source routing. While connect-time failover and client load balancing choose an address from a list, source routing connects to each address in the list sequentially.

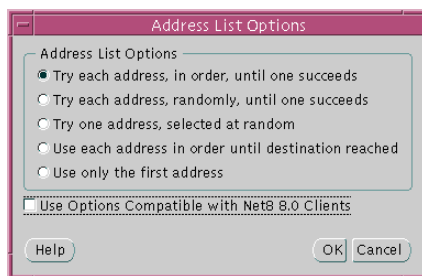
To enable or disable client load balancing or connect-time failover, use the Net8 Assistant.

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Double-click the Net Service Names folder in the directory tree.
3. Select the net service name.



4. Click the Advanced in the Address Configuration group.

The *Address List Options* dialog box appears:



5. Select any of the address list options, then click OK.
6. Choose Save Network Configuration from the File menu.
The TNSNAMES.ORA file is re-created.
7. Choose Exit from the File menu.

The Net8 Assistant application exits.

In the example below, a TNSNAMES.ORA is configured for client load balancing with the *Try each address, randomly, until one succeeds* option:

```
sales=
  (description=
    (load_balance=on)
    (address=(protocol=tcp)(host=sales-pc)(port=1521)
    )
    (address=(protocol=tcp)(host=hr-pc)(port=1521)
    )
    (connect_data=
      (service_name=sales.com))
  )
```

In the example below, a TNSNAMES.ORA is configured for connect-time failover with the *Try each address, in order, until one succeeds* option:

```
sales=
  (description=
    (address_list=
      (load_balance=off)
      (failover=on)
      (address=(protocol=tcp)(host=hr-pc)(port=1521)
      )
      (address=(protocol=tcp)(host=finance-pc)(port=1521)
      )
    )
    (connect_data=(service_name=sales.com))
  )
```


Configuring a Profile

A profile on the client configures functionality and defines how Net8 works to establish and maintain connections with services on the network. A profile is stored and implemented through a configuration file called `SQLNET.ORA`, located in `$ORACLE_HOME/network/admin` on UNIX and `ORACLE_HOME\network\admin` on Windows platforms. A profile allows configuration of:

- naming method prioritization
- tracing and logging features
- connections through specific processes
- security features

This section describes the following profile features:

- [Configuring Domains](#)
- [Prioritizing Net Service Name Naming Method Resolution](#)
- [Configuring Preferred Oracle Names Servers](#)
- [Configuring Advanced Profile Information](#)
- [Routing Connection Requests](#)
- [Configuring Oracle Advanced Security](#)

For a complete list of all available parameters that may be configured in a profile, see "[Profile Parameters \(SQLNET.ORA\)](#)" on page C-7.

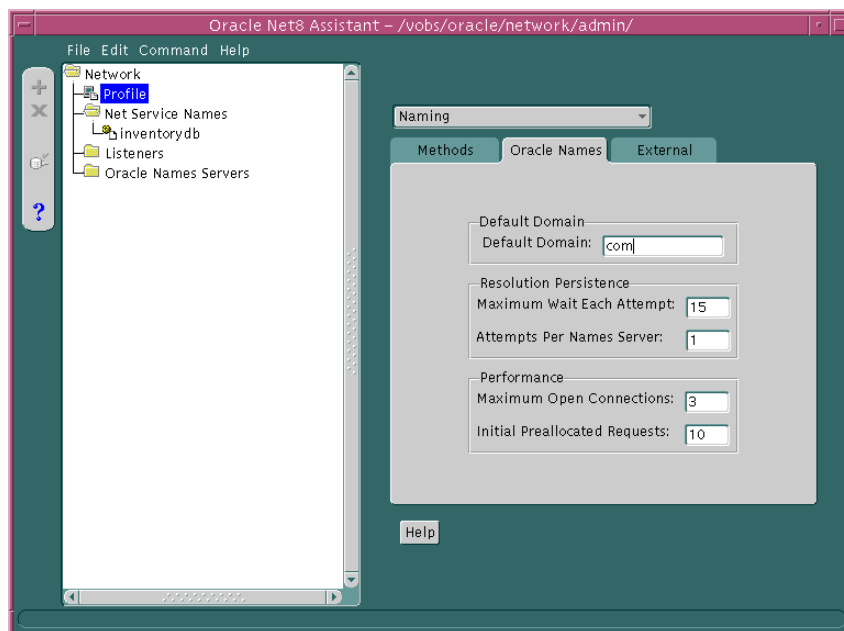
Configuring Domains

The `NAMES.DEFAULT_DOMAIN` parameter in `SQLNET.ORA` defines the domain from which the client most often requests names. When this parameter is set, the domain name is automatically appended to the net service name in a connect string. If this parameter is set with the Net8 Assistant, the domain is appended to any new net service name created without a domain. For example, if `NAMES.DEFAULT_DOMAIN=COM` is set, the connect string `SCOTT/TIGER@ SALES` gets searched in the `TNSNAMES.ORA` as `SALES.COM`. If the connect string includes the domain extension, (such as `SCOTT/TIGER@SALES.COM`), the domain is not appended.

To specify a domain, use the Net8 Assistant.

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Click the Profile icon.
3. Choose Naming from the drop-down list box.

4. Click the Oracle Names tab:



5. Enter the domain in the Default Domain field.
6. Choose Save Network Configuration from the File menu.
7. Choose Exit from the File menu to exit the Net8 Assistant application.

The SQLNET.ORA should contain an entry that looks like the following:

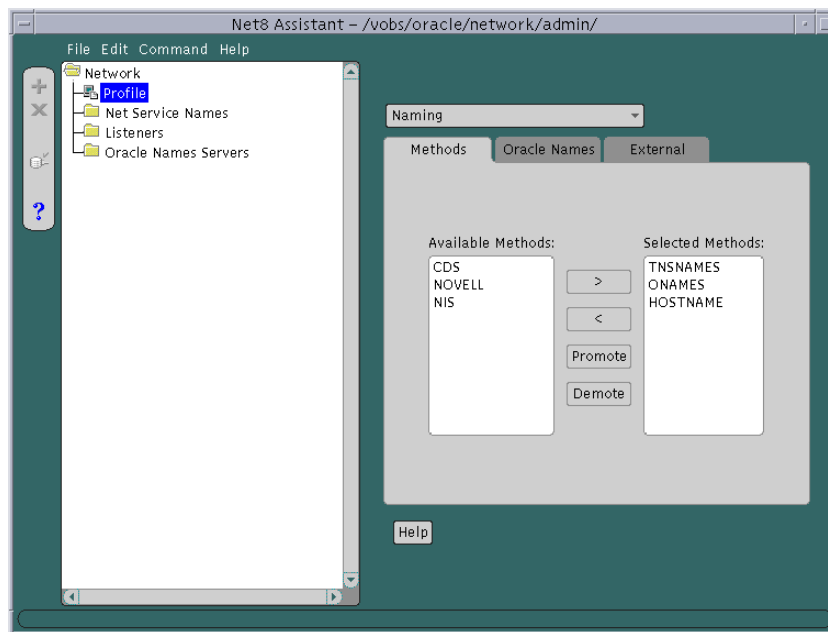
```
names.default_domain=com
```

Prioritizing Net Service Name Naming Method Resolution

After naming methods are configured, as documented in [Chapter 6, "Configuring Naming Methods and the Listener"](#), they must be prioritized. The naming method at the top of the list is used first to resolve a net service name. If it cannot resolve the net service name, the second method in the list is used to resolve the net service name.

To specify the order of naming methods, use the Net8 Assistant.

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Click the Profile icon.
3. Choose Naming from the drop-down list box.
4. Click the Methods tab:



5. Choose from the Available Methods list which naming method you want to use.

6. Order the naming methods according to which method you want Net8 to try first when resolving net service names. Select the naming method in the Selected Methods list, and use the Promote or Demote button to move the selection up or down in the list.
7. Choose Save Network Configuration from the File menu.
8. Choose Exit from the File menu to exit the Net8 Assistant application.

The SQLNET.ORA should contain an entry that looks like the following:

```
names.directory_path=(onames, tnsnames)
```

Available naming methods are described below:

Naming Method	Description
ONAMES	Oracle Names Net service names are resolved centrally, through an Oracle Names server.
TNSNAMES	Local Naming Net service names are resolved using the TNSNAMES.ORA file which resides on the client.
HOSTNAME	Host Naming Net service names are resolved using the host naming method. Certain criteria must be met to use host naming names resolution.
CDS	Cell Directory Services This naming method is available with Oracle Advanced Security option. Additional Information: See <i>Oracle Advanced Security Administrator's Guide</i> .
NDS	NetWare Directory Service (NDS)
NIS	Network Information Service (NIS)

Configuring Preferred Oracle Names Servers

A [preferred Oracle Names Server](#) is an Oracle Names Server that is preferred by a client for names resolution. It is normally the Oracle Names server that is physically closest to the client, or available over the least expensive network link. Most importantly, a preferred Oracle Names server should be able to resolve the net service names of the services you use most frequently.

If you are using Oracle Names as a naming service (though not necessarily running an Oracle Names Server on your local machine), you can specify the name and addresses of preferred Oracle Names servers to take precedence over any other available Oracle Names server address. Net8 will route connect requests to each preferred Names Server until a response is received.

Note: Preferred Oracle Names servers are an alternative to using the Discover Oracle Names Servers option in the Net8 Assistant or the REORDER_NS command in the NAMESCTL control utility, a newer feature that creates a list of Oracle Names servers based on what is available throughout the network, and ranks them in the order of fastest response time. Preferred Oracle Names servers override any other Oracle Names servers found in the discovery process. Once you have initially discovered a Names Server and generated a Names Server list file, you may want to delete preferred Oracle Names Servers.

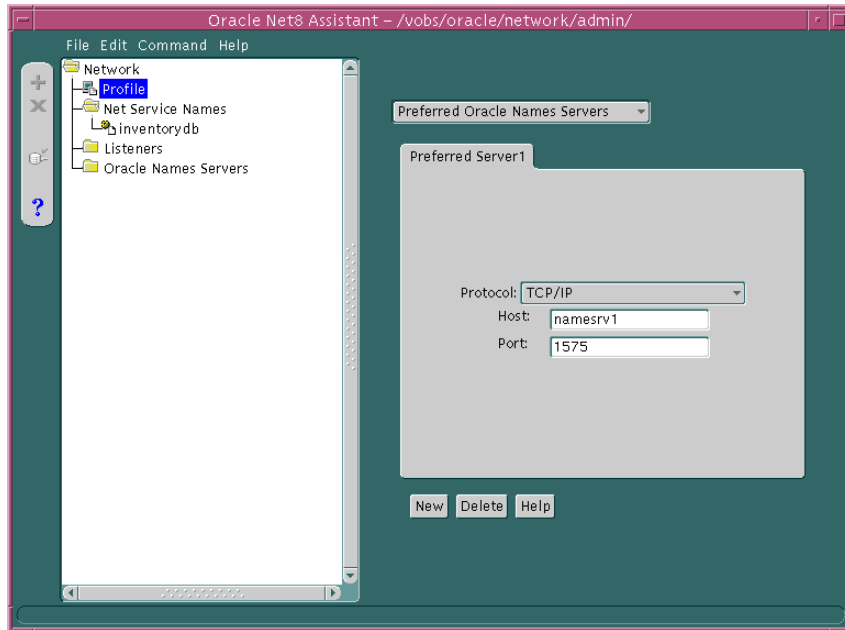
See "[Differences Between Versions of Oracle Names](#)" on page 2-42 for a description of how Oracle Names server lists are created.

The `NAMES.PREFERRED_SERVERS` parameter in `SQLNET.ORA` specifies the preferred Oracle Names server(s). If you configure preferred Oracle Names server(s), the Oracle Names listening protocol address(es) you configured in `NAMES.ORA` must match the address(es) you configure with the `NAMES.PREFERRED_SERVERS` in `SQLNET.ORA`.

To specify a preferred Oracle Names server:

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose `Start > Programs > Oracle - HOME_NAME > Network Administration > Net8 Assistant`.
2. Click the Profile icon.
3. Choose Preferred Oracle Names Servers from the drop-down list box:

4. Click New:



5. Choose the protocol and enter the requested protocol address information for an Oracle Names server configured on that address, as described on "[Configuring Protocol Addresses](#)" on page 6-76.
6. Choose Save Network Configuration from the File menu.
7. Choose Exit from the File menu to exit the Net8 Assistant application.

The SQLNET.ORA should contain an entry that looks like the following:

```
names.preferred_servers=  
  (address= (protocol=tcp) (host=namesrv1) (port=1575))
```


Configuring Advanced Profile Information

The profile configuration file `SQLNET.ORA` can be configured with the following advanced features:

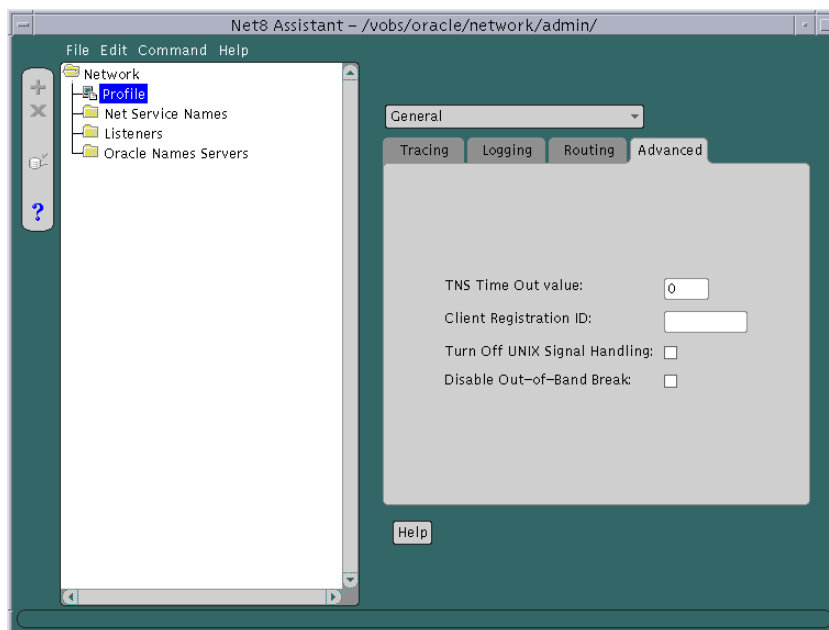
Net8 Assistant option	SQLNET.ORA Parameter	Description
TNS Time Out Value	<code>SQLNET.EXPIRE_TIME</code>	<p>This option can only be used on the server.</p> <p>Net8 sends a probe periodically to verify that a client-server connection is still active. This is done to ensure that connections are not left open indefinitely, due to an abnormal client termination. If the probe finds a dead connection, or a connection that is no longer in use, it returns an error, causing the server process to exit.</p> <p>Limitations on using the dead connection detection feature are as follows:</p> <ul style="list-style-type: none"> ■ Dead connection detection is not allowed on bequeathed connections. ■ Though very small, a probe packet generates additional traffic that may downgrade network performance. ■ The server may need to perform additional processing to distinguish the connection probing event from other events that occur, depending on which operating system is in use. This may also result in downgrading network performance.
Client Registration ID	<code>SQLNET.CLIENT_REGISTRATION</code>	Sets a unique identifier for this client machine. This identifier will be passed to the listener with any connection request. The identifier can be any string up to 128 characters long.

Net8 Assistant option	SQLNET.ORA Parameter	Description
Turn Off UNIX Signal Handling	BEQUEATH_DETACH	Since the client application spawns a server process internally through the Bequeath protocol as a child process, the client application becomes responsible for cleaning up the child process when it completes. When the server process completes its connection responsibilities, it becomes a defunct process. Signal handlers are responsible for cleaning up these defunct processes. Setting this parameter configures the client profile to pass this process to the UNIX init process by disabling signal handlers.
Disable Out-of-Band Break	DISABLE_OOB	Disables out of band breaks if the underlying protocol does not support this feature.

To set advanced features for clients, use either or the Net8 Assistant:

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Click the Profile icon.
3. Choose General from the drop-down list box.

4. Click the Advanced tab:



5. Enter the values for the option(s) you would like to set.
6. Choose Save Network Configuration from the File menu.
7. Choose Exit from the File menu to exit the Net8 Assistant application.

Routing Connection Requests

The client and servers acting as clients can be configured so connection requests are directed to a specific process at *all* times. The following options in the Net8 Assistant or parameters in SQLNET.ORA file may be specified:

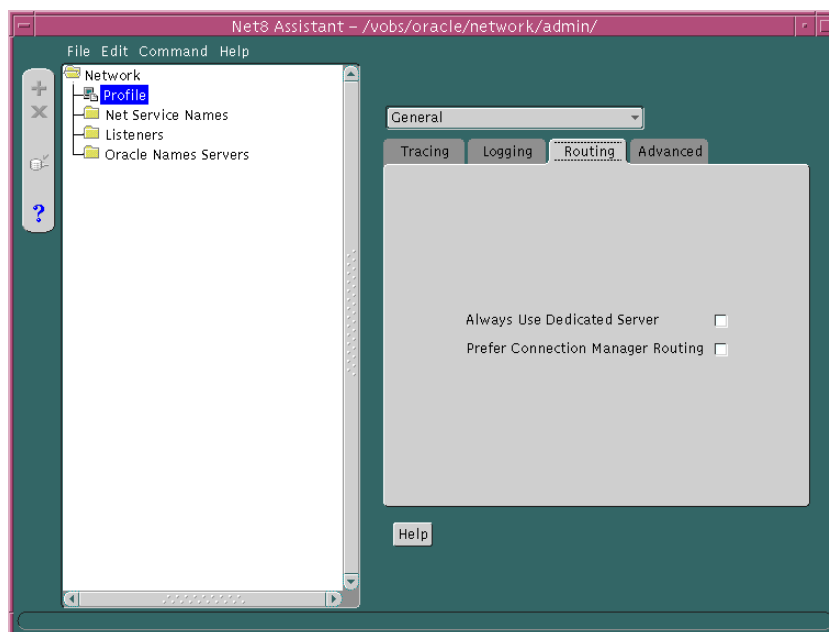
Net8 Assistant option	SQLNET.ORA Parameter	Description
Always Use Dedicated Server	USE_DEDICATED_SERVER	Forces the listener to spawn a dedicated server for all network session from this client.
Prefer Connection Manager Routing	USE_CMAN	Instructs the client to try to use an Oracle Connection Manager whenever possible. If no Oracle Connection Manager addresses are available, connections are routed through any available listener address. Note: If you are using Oracle Connection Manager with Oracle Names, this option must be set on clients and Oracle Names servers.

Note: While the Net8 Assistant displays the Use IPC Address for Client, this feature is not enabled because it's corresponding parameter, AUTOMATIC_IPC, has been disabled for release 8.1.

To route connection requests, use the Net8 Assistant:

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Click the Profile icon.
3. Choose General from the drop-down list box.

4. Click the Routing tab:



5. Click the preferred way you want connection requests routed.
6. Choose Save Network Configuration from the File menu.
7. Choose Exit from the File menu to exit the Net8 Assistant application.

Configuring Oracle Advanced Security

Oracle Advanced Security is an optional product that works with Net8 and SQL*Net release 2.1.4 and later. It includes the following features:

- [Security Services](#)
- [Authentication Services](#)
- [Oracle DCE Integration](#)

Security Services

Oracle Advanced Security enables Net8 and related products to use network data encryption and checksumming so that data cannot be read or altered. It protects data from unauthorized viewing by using the RSA Data Security RC4 or the Data Encryption Standard (DES) encryption algorithm. To ensure that data has not been modified, deleted, or replayed during transmission, the security services of Oracle Advanced Security can generate a cryptographically secure message digest and include it with each packet sent across the network.

Oracle Advanced Security is supported by Oracle Connection Manager. Clients and servers using different protocols can securely transfer data across network protocol boundaries. For example, clients using LAN protocols such as NetWare (SPX) can share data securely with large servers using different protocols such as LU6.2, TCP/IP, or DECnet.

Authentication Services

Oracle Advanced Security includes enhanced user authentication services such as support for single sign-on. These authentication services enhance the existing security facilities of Oracle7 and Oracle8i such as secure network access control log on, roles, and auditing by providing reliable user identification. No changes to applications are required. Oracle Advanced Security works over all protocols, operating systems, and name services. It also supports token authentication through Security Dynamics ACE Server, Kerberos, RADIUS protocol or RADIUS compatible server, Secure Sockets Layer (SSL), DCE Security Server, and biometrics authentication through Identix.

These services are available to most products that implement Net8, including the Oracle8i, Developer 2000 tools, and any other Oracle or third-party products that support Net8.

Oracle DCE Integration

Oracle Distributed Computing Environment (DCE) Integration is an optional product that works with Net8 and SQL*Net 2.1.6 and later. It enables users to transparently use Oracle tools and applications to access Oracle7 and Oracle8i servers in a DCE environment. It provides authenticated RPC (Remote Procedure Call) as the transport mechanism, which enables multi-vendor interoperability. The DCE security service enables a user logged onto DCE to securely access any Oracle application without having to specify a user name or password. This is sometimes referred to as "external authentication", formerly referred to as "OPSS support".

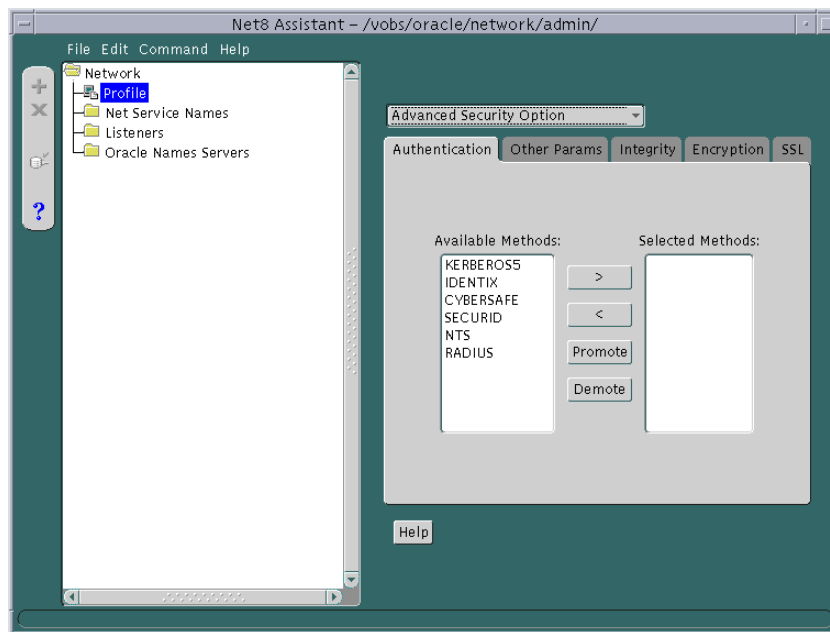
Oracle DCE Integration also provides support for DCE Cell Directory Service (CDS), which allows Oracle7 and Oracle8i services to be transparently accessed throughout the DCE environment. Users can connect to Oracle database servers in a DCE environment using familiar Oracle service names. Oracle service names can be managed from a central location with standard DCE tools. For more information, see *Oracle Advanced Security Administrator's Guide* and your Oracle platform-specific documentation.

Configuration

To configure a client to use Advanced Security option:

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Click the Profile icon.

3. Choose Advanced Security Option from the drop-down list box:



See *Oracle Advanced Security Administrator's Guide* for further information about configuration.

4. Choose or edit options as applicable.
5. Choose Save Network Configuration from the File menu.
6. Choose Exit from the File menu to exit the Net8 Assistant application.

Enabling Connection Concentration

Oracle Connection Manager enables you to take advantage of Net8's ability to multiplex or funnel multiple logical client network sessions through a single transport connection to a multi-threaded server destination. This is accomplished through Oracle Connection Manager's connection concentration feature.

Step 1: Configure Oracle Connection Manager

The gateway process, CMGW, accepts connection requests at the following default listening address: "[Oracle Connection Manager Processes](#)"

```
cmgw=(address=(protocol=tcp)(host=anyhost)(port=1630))
```

The administrative process, CMADMIN, listens for local and remote administration commands at the following listening address:

```
cmadmin=(address=(protocol=tcp)(host=anyhost)(port=1830))
```

If you do not want to use TCP/IP or the ports 1630 and 1830, you must create a CMAN.ORA with the following information, if one does not exist with the address information, at `$ORACLE_HOME/network/admin` on UNIX and `ORACLE_HOME\network\admin` on Windows NT:

```
cmgw=(address=(protocol_address_information))
cmadmin=(address=(protocol_address_information))
```

The Net8 Assistant does not support configuration of CMAN.ORA, so changes must be made manually.

See also the following:

- "[Oracle Connection Manager Processes](#)" on page 2-46 for further information about CMGW and CMADMIN
- "[Configuring Protocol Addresses](#)" on page 6-76 for information about proper protocol address parameter usage

Step 2: Configure the Server

Verify that the destination server is configured as a multi-threaded server and that the multiplexing feature is turned on. This is done by setting the `MTS_DISPATCHERS` parameter in the database initialization file (`INITSID.ORA`) for that instance with the `PROTOCOL` and `MULIPIXEX` attribute.

For example:

```
mts_dispatchers="(protocol=tcp) (multiplex=on)"
```

Attribute	Description
<code>PROTOCOL</code> (<code>PRO</code> or <code>PROT</code>)	The network protocol (TCP in the example above) for which the dispatcher will generate a listening end point.
<code>MULTIPIXEX</code> (<code>MUL</code> or <code>MULT</code>)	Used to enable connection concentration. If 1, ON, YES, TRUE, or BOTH is specified, then connection concentration is enabled for both incoming and outgoing network connections. If IN is specified, the connection concentration is enabled for incoming network connections from the client. If OUT is specified, the connection concentration is enabled for outgoing network connections. If 0, NO, OFF, or FALSE is specified, then connection concentration is disabled for both incoming and outgoing network connections.

For more information about configuring MTS, see [Chapter 9, "Configuring Multi-Threaded Server"](#).

Step 3: Configure the Client

Client support is accomplished by setting the Oracle Connection Manager address and listener address through a TNSNAMES.ORA file or an Oracle Names server:

- If using a TNSNAMES.ORA file, the Oracle Connection Manager address and the SOURCE_ROUTE parameter must be set, which creates a source route of addresses through an Oracle Connection Manager to the destination database. To configure the local naming method's TNSNAMES.ORA file, follow the instructions in "[Configuring the Client for Oracle Connection Manager Features](#)" of this chapter.
- If using Oracle Names servers, the Oracle Connection Manager automatically updates the addresses in the Oracle Names servers, inserting the address for the Oracle Connection Manager into the existing addresses. In order for Oracle Names servers to send requests to Oracle Connection Manager, USE_CMAN=TRUE must be set in the SQLNET.ORA file on clients and machines where each Oracle Names server resides. To set this parameter, follow the steps in "[Routing Connection Requests](#)" on page 7-26.

Figure 7-1 shows a comparison of a regular TNSNAMES.ORA file and a TNSNAMES.ORA file with an entry to use the Oracle Connection Manager (Connection Manager entries are shown in boldface text):

Figure 7-1 TNSNAMES.ORA File with and without Oracle Connection Manager

```

sales=                                sales=
  (description=                        (description=
    (address=                          ( source_route=yes)
      (protocol=tcp)                  ( address=
      (host=sales-pc)                 ( protocol=tcp)
      (port=1521)                     ( host=cman-pc)
    )                                  ( port=1630)
  )                                    )
  (connect_data=                      )
    (service_name=sales,com)          (address=
  )                                    (protocol=tcp)
)                                       (host=sales-pc)
                                       (port=1521)
                                       )
                                       (connect_data=
                                       (service_name=sales.com)
                                       )
                                       )

```

TNSNAMES.ORA Element	Description
<code>source_route=yes</code>	Creates a source route of addresses through an Oracle Connection Manager to the destination database.
<code>(address= (protocol=tcp) (host=cman-pc) (port=1630))</code>	This first address is to the Oracle Connection Manager. From here, the Oracle Connection Manager connects to the service through the listener.

Enabling Multi-Protocol Support

Note: If more than one Oracle Connection Manager is used in the connection path (more than 1 hop), you cannot use Oracle Names to connect clients through it.

Oracle Connection Manager also provides multiple protocol support enabling a client and server with different networking protocols to communicate with each other. An Oracle Connection Manager can listen on any protocol that Oracle supports.

Without multi-protocol support, a client that uses SPX cannot connect to a server that uses TCP/IP. If Oracle Connection Manager is configured for TCP/IP, the client can connect to Oracle Connection Manager using SPX, and Oracle Connection Manager can connect to the server using TCP/IP.

Step 1: Configure Oracle Connection Manager

The gateway process, CMGW, accepts connection requests at the following default listening address:

```
cman=(address=(protocol=tcp)(host=anyhost)(port=1630))
```

The administrative process, CMADMIN, listens for local and remote administration commands at the following listening address:

```
cman_admin=(address=(protocol=tcp)(host=anyhost)(port=1830))
```

If you do not want to use TCP/IP or the ports 1630 and 1830, you must create a CMAN.ORA with the following information, if one does not exist with the address information, at `$ORACLE_HOME/network/admin` on UNIX and `ORACLE_HOME\network\admin` on Windows NT:

```
cman=(address=(protocol_address_information))
cman_admin=(address=(protocol_address_information))
```

The Net8 Assistant does not support configuration of CMAN.ORA, so changes must be made manually.

See also the following:

- ["Oracle Connection Manager Processes"](#) on page 2-46 for further information about CMGW and CMADMIN
- ["Configuring Protocol Addresses"](#) on page 6-76 for information about proper protocol address parameter usage

Step 2: Configure the Client

Multi-protocol support is accomplished by setting the Oracle Connection Manager address and listener address through a TNSNAMES.ORA file or Oracle Names servers:

- If using a TNSNAMES.ORA file, the Oracle Connection Manager address and the SOURCE_ROUTE parameter must be set, which creates a source route of addresses through an Oracle Connection Manager to the destination database. To configure the local naming method's TNSNAMES.ORA file, follow the instructions in "[Configuring the Client for Oracle Connection Manager Features](#)" of this chapter and these tips:
 - Configure the client first with the address of the Oracle Connection Manager, using a protocol available on the client.
 - Configure the client with the address of the listener, using a protocol available on the server.
 - The Oracle Connection Manager machine must contain all the protocols used on the clients and server machines.
- If using Oracle Names servers, the Oracle Connection Manager automatically updates the addresses in the Oracle Names servers, inserting the address for the Oracle Connection Manager into the existing addresses. In order for Oracle Names servers to send requests to Oracle Connection Manager, USE_CMAN=TRUE must be set in the SQLNET.ORA file on clients and machines where each Oracle Names server resides. To set this parameter, follow the steps in "[Routing Connection Requests](#)" on page 7-26.

Figure 7-2 depicts a client using SPX to connect to an Oracle Connection Manager, and Oracle Connection Manager using TCP/IP to connect to a database server.

Figure 7-2 CMAN Parameter Configured to Use Multi-Protocol Support

```

cman=
  (description=
    (source_route=yes)
    (address=
      (protocol=spx)
      (service=cman)
    )
    (address=
      (protocol=tcp)
      (host=sales-pc)
      (port=1521)
    )
    (connect_data=
      (service_name=sales.com)
    )
  )

```

TNSNAMES.ORA Element	Description
source_route=yes	Creates a source route of addresses through an Oracle Connection Manager to the destination database.
(address= (protocol=spx) (service=cman))	This first address is from a client using SPX to the Oracle Connection Manager. From here, the Oracle Connection Manager connects to the service using TCP/IP through the listener.

Enabling Net8 Access Control

Oracle Connection Manager also includes a feature which you can use to control client access to designated servers in a TCP/IP environment. By specifying certain filtering rules, you may allow or restrict specific clients access to a server.

This feature requires the release 8.1 Oracle Connection Manager if there are release 8.1 services in the network.

Step 1: Configure Oracle Connection Manager

To configure the Oracle Connection Manager:

1. Create a CMAN.ORA file, if one does not already exist, in `$ORACLE_HOME/network/admin` on UNIX and `ORACLE_HOME\network\admin` on Windows NT. The Net8 Assistant does not support configuration of CMAN.ORA.
2. If you do not want to use the default listening port 1630 for the CMGW process, add the following:

```
cmgw=(address=(protocol=tcp)(host=anyhost)(port=port))
```

See "[CMGW](#)" on page 2-46 for further information about CMGW.

3. If you do not want to use the default listening port 1830 for the CMADMIN process, add the following:

```
cmadmin=(address=(protocol=tcp)(host=anyhost)(port=port))
```

See "[CMADMIN](#)" on page 2-47 for further information about CMGW.

4. Add a CMAN_RULES section with the following parameters:

```

cman_rules=
  (rule_list=
    (rule=
      (src=shost)
      (dst=dhost)
      (srv=service)
      (act=accept | reject)
    )
  )

```

Parameter	Description
<i>SHOST</i>	source host name or IP address (client)
<i>DHOST</i>	destination host name or IP address (server)
<i>SERVICE</i>	SERVICE_NAME (obtained from the INITSID.ORA file) value of release 8.1 service or SID pre-8.1 service
<i>ACCEPT REJECT</i>	accept or reject the incoming requests based on the above three parameters.

Multiple RULEs can be defined within the RULE_LIST. The action (ACT) in the first matched RULE is applied to the request. When CMAN_RULES exists, the Oracle Connection Manager adheres to the principle *that which is not expressly permitted is prohibited*. If the CMAN_RULES are not defined, then everything is permitted.

The example below shows restriction to service SALES.COM for a client machine of CLIENT1-PC, and access to service DB1 for client 144.25.23.45.

```

cman_rules=
  (rule_list=
    (rule=(src=client1-pc)(dst=sales-pc)(srv=sales.com)(act=reject))
    (rule=(src=144.25.23.45)(dst=144.25.187.200)(srv=db1)(act=accept))
  )

```

Step 2: Configure the Client

Client support is accomplished by setting the Oracle Connection Manager address and listener address through a TNSNAMES.ORA file or an Oracle Names server:

- If using a TNSNAMES.ORA file, the Oracle Connection Manager address and the SOURCE_ROUTE parameter must be set, which creates a source route of addresses through an Oracle Connection Manager to the destination database. To configure the local naming method's TNSNAMES.ORA file, follow the instructions in "[Configuring the Client for Oracle Connection Manager Features](#)" of this chapter and these tips:
 - Configure the client first with the address of the Oracle Connection Manager, using only TCP/IP.
 - Configure the client with the address of the listener, using a protocol available on the Oracle Connection Manager machine.
 - Configure the client with SOURCE_ROUTE=YES to creates a source route of addresses through a Connection Managers to the destination database.
- If using Oracle Names servers, the Oracle Connection Manager automatically updates the addresses in the Oracle Names servers, inserting the address for the Oracle Connection Manager into the existing addresses. In order for Oracle Names servers to send requests to Oracle Connection Manager, USE_CMAN=TRUE must be set in the SQLNET.ORA file on clients and machines where each Oracle Names server resides. To set this parameter, follow the steps in "[Routing Connection Requests](#)" on page 7-26.

Configuring the Client for Oracle Connection Manager Features

Note: Oracle Corporation recommends not setting client load balancing or connect-time failover when using the Oracle Connection Manager, as the address list order has a specific order and meaning. When SOURCE_ROUTE is set, the first address is for the client connection to an Oracle Connection Manager and the second address is for the Oracle Connection Manager connection to a listener. Client load balancing and connect-time failover are disabled when SOURCE_ROUTE is set.

Note: If more than one Oracle Connection Manager is used in the connection path, you cannot use Oracle Names to connect clients through it.

Configuring the client involves routing client connection requests you want concentrated to the database server through that a Connection Manager. This is achieved by setting the Oracle Connection Manager address through a TNSNAMES.ORA file or an Oracle Names server:

- If using Oracle Names servers, the Oracle Connection Manager automatically updates the addresses in the Oracle Names servers, inserting the address for the Oracle Connection Manager into the existing addresses. In order for Oracle Names servers to send requests to Oracle Connection Manager, USE_CMAN=TRUE must be set in the SQLNET.ORA file on machines where each Oracle Names server resides. To set this parameter, follow the steps in "[Routing Connection Requests](#)" on page 7-26.
- If using a TNSNAMES.ORA file, the Oracle Connection Manager address and the SOURCE_ROUTE parameter must be set, which creates a source route of addresses through an Oracle Connection Manager to the destination database. To configure the local naming method's TNSNAMES.ORA file, follow these steps:

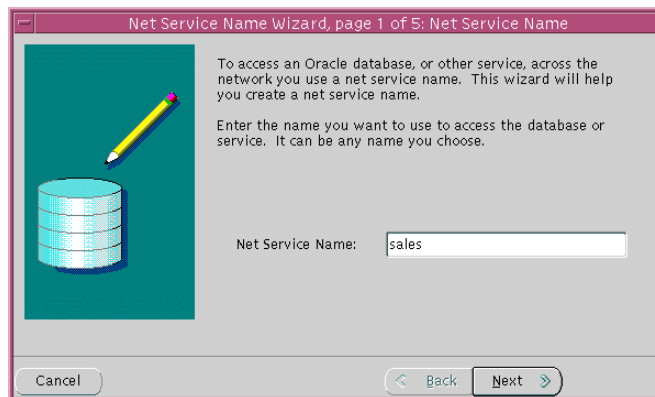
[Step 1: Configure an Oracle Connection Manager Address](#)

[Step 2: Configure the Listener Address](#)

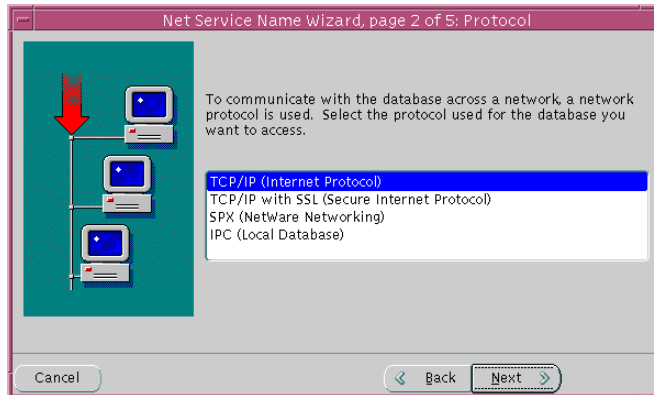
Step 1: Configure an Oracle Connection Manager Address

To configure an Oracle Connection Manager address in TNSNAMES.ORA:

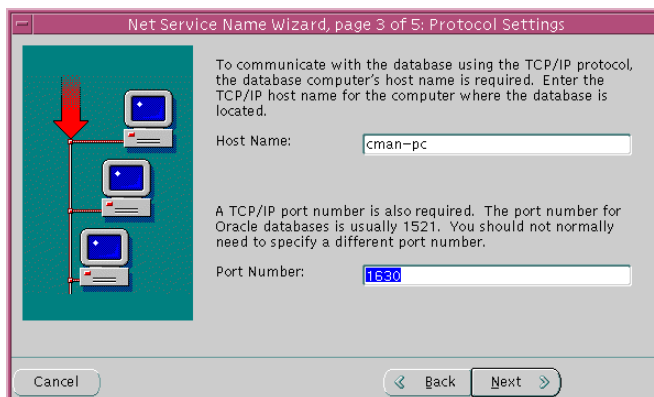
1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Double-click on the Net Service Names folder in the directory tree.
3. Select Create from the Edit menu or click on the "+" button to create a new net service name. The *Net Service Name Wizard* starts.
4. Enter any name in the Net Service Name field, then click Next:



5. Select the protocol the CMGW process is configured to listen on, then click Next. By default this protocol is TCP/IP.

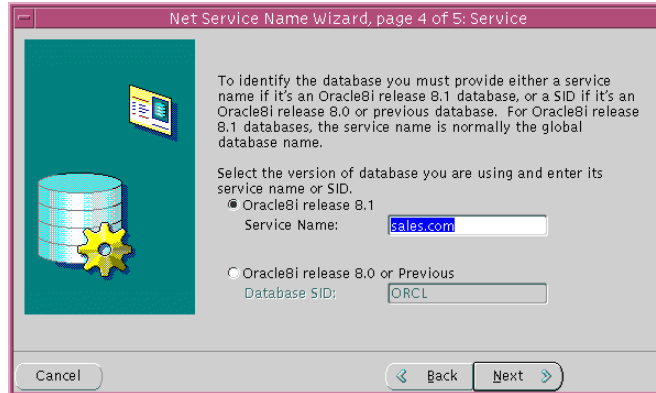


6. Enter network protocol information for the Oracle Connection Manager, then click Next. If you are using TCP/IP, the default port for UNIX operating systems is 1630.



For further information about protocol parameters, see "[Configuring Protocol Addresses](#)" on page 6-76.

7. If the destination database is release 8.1, click Oracle8 Version 8.1, enter a service name in the Service Name field, then click Next:



The service name is typically the global database name. See "[Setting Service Names and Instance Names](#)" on page 6-3 for further information about the service name string to use.

If destination service is prior to release 8.1, click Oracle8 Version 8.0 or Previous, enter a SID in the Database SID field, then click Next.

8. Click Next.

Note: Do *not* click Test, as the connection will fail at this point.

9. Click Finish to save your configuration and dismiss *Net Service Name Wizard*.

The new net service name and the Oracle Connection Manager address is added to the Net Service Names folder and the TNSNAMES.ORA.

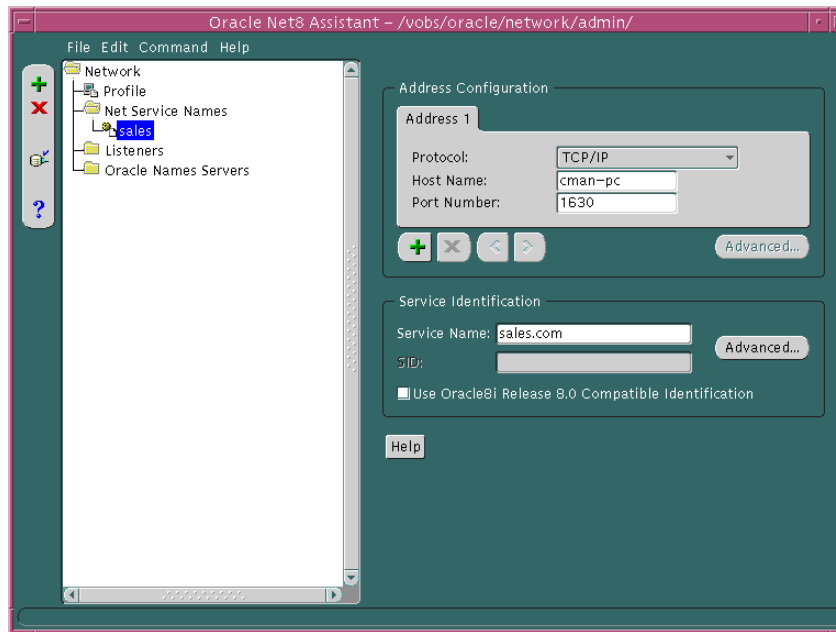
Step 2: Configure the Listener Address

After the Oracle Connection Manager address is specified, create an address for the listener, so the Oracle Connection Manager can connect to the server.

To configure the listener address in TNSNAMES.ORA:

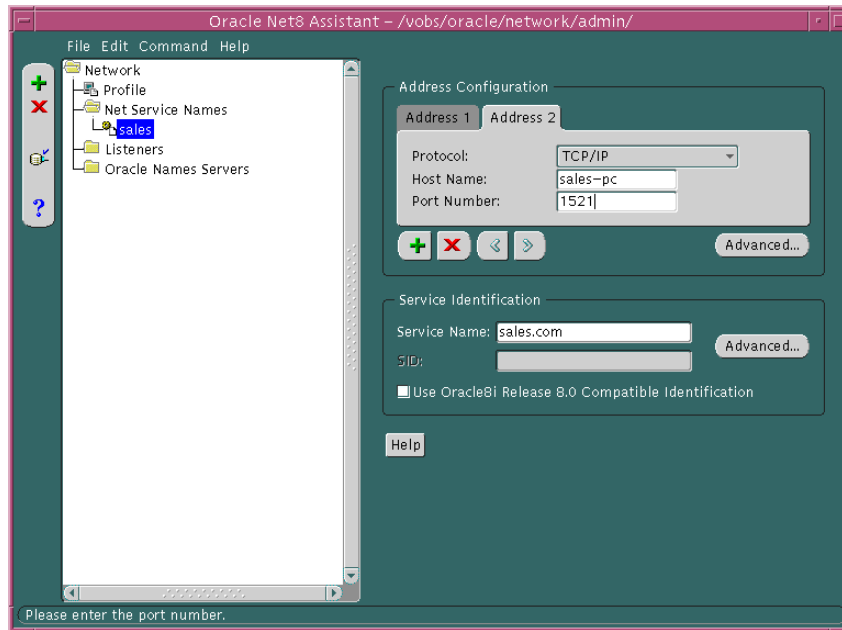
1. Double-click the Net Service Names folder.
2. Click on the net service name you created in "Step 1: Configure an Oracle Connection Manager Address" on page 7-43.

Notice the Oracle Connection Manager address is displayed in the Address 1 tab:



3. Click "+".
A new address tab displays.

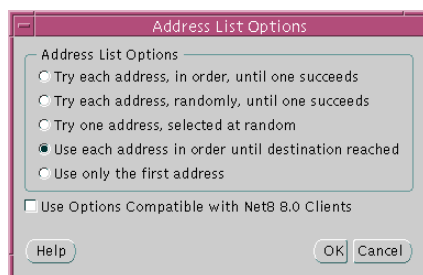
4. Select the protocol, then enter protocol information for the listener. If you are using TCP/IP, the default port number is 1521:



For further information about protocol parameters, see "[Configuring Protocol Addresses](#)" on page 6-76.

5. Click Advanced in the Address Configuration group box.

The *Address List Options* dialog box appears:



6. Click *Use each address in order until destination reached* to ON.

This option creates tells the client to connect to the first address, the Oracle Connection Manager, and from the first address to the second address, the listener.

7. Click OK to dismiss the dialog box.
8. Choose Save Network Configuration from the File menu.
9. Choose Exit from the File menu to exit the Net8 Assistant.

The listener address is added to the TNSNAMES.ORA file.

Enabling Connection Pooling

Connection pooling is a resource utilization feature that allows you to maximize the number of physical network connections to a multi-threaded server. This is achieved by sharing or pooling a dispatcher's set of connections among multiple client processes.

Verify that the destination server is configured as a multi-threaded server and that the connection pooling parameters are set. This is accomplished by setting the `MTS_DISPATCHERS` parameter in the database initialization file (`INITSID.ORA`) with the `PROTOCOL`, `DISPATCHERS`, `POOL`, and `CONNECTIONS` attributes.

For example:

```
mts_dispatchers="(protocol=tcp) (dispatchers=2) (pool=yes) (connections=2) "
```

Attribute	Description
CONNECTIONS (CON or CONN)	The maximum number of network connections to allow for each dispatcher. The default is 1022.
DISPATCHERS (DIS or DISP)	The initial number of dispatcher to start. The default is 1.
PROTOCOL (PRO or PROT)	The network protocol (TCP in the example above) for which the dispatcher will generate a listening end point.
POOL	<p>If a number is specified, then connection pooling is enabled for both incoming and outgoing network connections and the number specified is the timeout in ticks for both incoming and outgoing network connections.</p> <p>If ON, YES, TRUE, or BOTH is specified, then connection pooling is enabled for both incoming and outgoing network connections and the default timeout of 10 will be used for both incoming and outgoing network connections.</p>
TICKS	The size of a network tick in seconds. The default is 15 seconds. This parameter is optional for connection pooling.

For more information about configuring MTS, see [Chapter 9, "Configuring Multi-Threaded Server"](#).

Configuring Java Option Connections

The Java option in Oracle8i can be programmed in three different ways:

- Java Stored Procedures — traditional database stored procedures, functions, triggers and object methods implemented in Java (or SQLJ)
- Enterprise Java Beans (EJBs) — standard server-oriented Java components
- CORBA Servers — standard CORBA 2.0 compliant servers implemented in Java.

EJBs and CORBA clients are invoked via the CORBA Internet Inter-Orb Protocol (IIOP) protocol.

Java Stored Procedures can run either in dedicated server mode or multi-threaded server (MTS) mode. EJBs and CORBA Servers run only in the MTS configuration. If you are primarily developing Java Stored Procedures, you may run them in the dedicated server configuration. If you are combining both EJBs and CORBA Servers along with Java stored procedures in a single application, you may configure your database to support EJBs and CORBA Servers in an MTS configuration and support stored procedures in dedicated server configuration.

If your network requires configuration of Java Stored Procedures in dedicated server mode and EJBs or CORBA Servers in MTS mode, see "[Configuring Both Multi-Threaded Server and Dedicated Server Modes](#)" on page 9-10.

This section covers the following topics:

- [Java Stored Procedure Configuration](#)
- [Enterprise JavaBeans and CORBA Configuration](#)
- [Using the Default Configuration of Oracle8i for EJBs and CORBA through the Listener](#)
- [Using a Non-Default Configuration](#)
- [Using SSL with EJBs and CORBA](#)

Java Stored Procedure Configuration

To configure the database to run Java Stored Procedures only, you must decide whether you want the database to run in dedicated server mode or MTS mode. If you are only planning to use Java stored procedures in dedicated server mode, you need to configure the database and clients, as described in the *Oracle8i Java Stored Procedures Developer's Guide*.

If you want to run Java Stored Procedures in MTS mode, you need to go through the steps associated with configuring the server for MTS mode with the MTS_DISPATCHERS parameter, as described in [Chapter 9, "Configuring Multi-Threaded Server"](#).

Enterprise JavaBeans and CORBA Configuration

In configuring Oracle8i for Enterprise JavaBeans and CORBA Servers, you need to be aware of the following issues:

- [IIOP Clients](#)
- [Session-Based IIOP and Configuration Issues](#)
- [Access through Listeners vs. Dispatchers \(Connection Load Balancing\)](#)
- [SSL Support with CORBA and EJB](#)

IIOP Clients

Clients access EJBs and CORBA Servers in the database via the Inter-Orb Protocol (IIOP) protocol. To support IIOP, the database must be configured in MTS mode with the General Inter-Orb Protocol (GIOP) presentation protocol. (IIOP is an implementation of GIOP over TCP/IP). Oracle8i provides a GIOP service implementation. The Oracle8i Java VM is a session-oriented Java VM. This means that each session in the database effectively gets its own VM as a private server. The Java VM, ORB, and database presentation layers are modeled to allow CORBA communication from multiple clients to EJBs and CORBA Servers running in the same or different sessions. For more information, see the *Oracle8i Enterprise JavaBeans and CORBA Developer's Guide*.

Session-Based IIOP and Configuration Issues

To support Oracle8i's session-oriented behavior, the GIOP facilities in the database have been structured to support two presentation protocols. Depending on which presentation you want to use, you need to choose the appropriate presentation handler:

- `oracle.aurora.server.SGiopServer` is provided for session-based IIOP connections. EJBs and CORBA are distinguished based on the session in which they live in addition to the standard IIOP semantics. Session-based IIOP is far more scalable than standard IIOP connections. Oracle Corporation recommends you use session-based IIOP connections. (Default)
- `oracle.aurora.server.GiopServer` for regular IIOP connections

To support IIOP, the appropriate presentation must be specified in the `PRESENTATION` attribute of the `MTS_DISPATCHERS` parameter. To handle session-based IIOP (which is the default behavior), `oracle.aurora.server.SGiopServer` is registered as the presentation handler in the `PRESENTATION` attribute of the `MTS_DISPATCHERS` parameter.

For more information about presentations and session IIOP, see the *Oracle8i Enterprise JavaBeans and CORBA Developer's Guide*.

Access through Listeners vs. Dispatchers (Connection Load Balancing)

Finally, EJB and CORBA clients that communicate with the database via IIOP can communicate with the database either by accessing a dispatcher directly or accessing it via a listener. There are benefits and trade-offs to both approaches.

- Access through a listener

For scalability, clients may connect to the Java option through the listener. The listener is configured to listen on a well-known port number, and the client communicates with the listener using this port number. To support CORBA and EJB, the listener must be configured to not just listen on a port for the TTC protocol but to also handle GIOP/and session-based GIOP connections. The listener load balances client requests among the dispatchers. While this configuration is more complex, it can also scale easily to support more than 1,000 concurrent connections.

- Access a dispatcher directly

The dispatcher is configured with the a well-known port number. The client uses this port number to connect directly to the dispatcher. This configuration has an easy setup, but scalability may become an issue if you need to support more than 1,000 concurrent connections.

Oracle Corporations recommends access through a listener.

SSL Support with CORBA and EJB

Oracle8i also supports the use of authentication data such as certificates and private keys required for use by SSL in combination with both types of GIOP protocols—regular GIOP and session-based GIOP. To use SSL with GIOP, you need to carry out two steps:

1. Configure the listeners and dispatchers to accept SSL by specifying TCPS as the protocol in the listener's address list.
2. Specify the SSL wallet to be used when configuring both the listener and dispatcher. For more information see "[Using SSL with EJBs and CORBA](#)".

Using the Default Configuration of Oracle8i for EJBs and CORBA through the Listener

During a Typical or Minimal installation of the server, or if you choose the Java VM Option using the Oracle Database Configuration Assistant, MTS will be configured automatically for session-based IIOP connections through the listener using TCP/IP.

The `INITSID.ORA` file is configured in the following manner:

```
mts_dispatchers="(protocol=tcp)(presentation=oracle.aurora.server.SGiopServer)"
```

The default attributes are described in the table below:

Parameter	Description
PROTOCOL (PRO or PROT)	Specifies the TCP/IP protocol, which the dispatcher will generate a listening end point.
PRESENTATION (PRE or PRES)	<p>Enables support of GIOP. Valid values for GIOP presentation values include:</p> <ul style="list-style-type: none"> ▪ oracle.aurora.server.SGiopServer for session-based GIOP dispatcher connections. This presentation is valid for TCP/IP and TCP/IP with SSL. ▪ oracle.aurora.server.GiopServer for standard GIOP dispatcher connections. This presentation is valid for TCP/IP and TCP/IP with SSL. <p>See the <i>Oracle8i Enterprise JavaBeans and CORBA Developer's Guide</i> for further information.</p>

The LISTENER.ORA file is configured in the following manner:

```
listener=
  (description_list=
    (description=
      (address_list=
        (address=(protocol=tcp)(host=sales-pc)(port=1521))
      )
    )
  )
  (description=
    (protocol_stack=
      (presentation=giop)
      (session=raw)
    )
    (address_list=
      (address=(protocol=tcp)(host=sales-pc)(port=2481))
    )
  )
)
```

LISTENER.ORA Element	Description
PROTOCOL_STACK	Identifies the presentation and session layer information for a connection.
(PRESENTATION=GIOP)	Identifies a presentation of GIOP for IIOP clients. GIOP supports both oracle.aurora.server.SGiopServer or oracle.aurora.server.GiopServer using TCP/IP.
(SESSION=RAW)	Identifies the session layer. There is no session-layer for IIOP clients.
(ADDRESS=...)	Specifies a listening address that uses TCP/IP on port 2481.

Using a Non-Default Configuration

If you are unable to use the default configuration, you can configure IIOP connections to the Java option through the listener or directly to the dispatcher with your own settings. This section covers the following topics:

- [Non-Default Configuration for Access through a Listener](#)
- [Configuring a Non-Default Listener](#)
- [Configuration to Access a Dispatcher Directly](#)

Non-Default Configuration for Access through a Listener

To configure IIOP client connections to the Java option through a listener, follow these steps:

[Step 1: Enable MTS and Configure IIOP in the INITSID.ORA File](#)

[Step 2: Configure the Listener](#)

[Step 3: Configure Clients](#)

Step 1: Enable MTS and Configure IOP in the INITSID.ORA File

MTS must be configured in the INITSID.ORA file in the following manner:

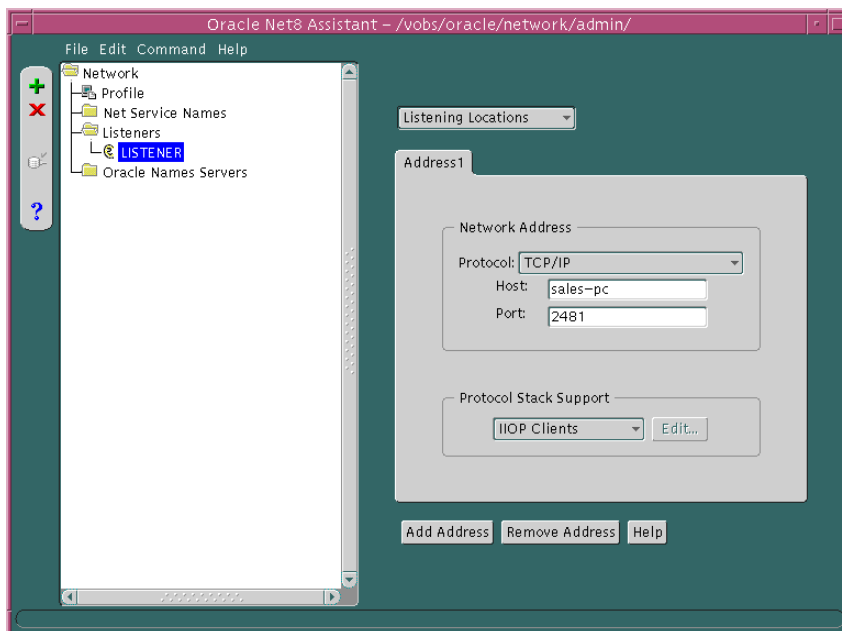
```
mts_dispatchers="(protocol=tcp | tcps)
(presentation=oracle.aurora.server.SGiopServer |
oracle.aurora.server.GiopServer)"
```

The attributes are described below:

Parameter	Description
PROTOCOL (PRO or PROT)	Specifies the TCP/IP or TCP/IP with SSL protocol, which the dispatcher will generate a listening end point for. Valid values: TCP (for TCP/IP) or TCPS (for TCP/IP with SSL)
PRESENTATION (PRE or PRES)	Enables support for GIOP. Valid values for GIOP presentation values include: <ul style="list-style-type: none"> ■ oracle.aurora.server.SGiopServer for session-based GIOP dispatcher connections. This presentation is valid for TCP/IP and TCP/IP with SSL. ■ oracle.aurora.server.GiopServer for standard GIOP dispatcher connections. This presentation is valid for TCP/IP and TCP/IP with SSL. See the <i>Oracle8i Enterprise JavaBeans and CORBA Developer's Guide</i> for further information.

Step 2: Configure the Listener Use the Net8 Assistant to modify any of the listener settings:

1. Start Net8 Assistant.
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Double-click on the Listeners folder.
3. Click on a listener.
4. Choose Listening Locations from the drop-down list box.



5. Choose TCP/IP or TCP/IP with SSL from the Protocol drop-down menu.
6. Enter the host name of the database in the Host field.
7. Enter port 2481 if chosen protocol is TCP/IP in the Port field, or enter port 2482 if the chosen protocol is TCP/IP with SSL in the Port field.

If you do not use port 2482 or 2483, follow the procedure in "[Configuring a Non-Default Listener](#)" on page 7-59.

8. Choose IIOP Client from the Client drop-down list box.
9. Choose Save Network Configuration from the File menu.
10. Choose Exit from the File menu to exit the Net8 Assistant.

Step 3: Configure Clients Use port number 2481 for TCP/IP or 2482 for TCP/IP with SSL for IIOP presentations. See the *Oracle8i Enterprise JavaBeans and CORBA Developer's Guide* documentation for further information.

Configuring a Non-Default Listener

In addition to the steps described in "[Non-Default Configuration for Access through a Listener](#)" on page 7-56, perform the following steps for a non-default listener:

- [Step 1: Configure Additional INITSID.ORA Configuration](#)
- [Step 2: Resolve the Listener Name](#)

Step 1: Configure Additional INITSID.ORA Configuration The non-default listener must be configured in the INITSID.ORA. How it is configured depends upon whether it is a local listener or a remote listener. Both scenarios are described in this section.

Local Listener

If you want the connection to go through a non-default *local* listener, INITSID.ORA *must* also be configured with the LOCAL_LISTENER parameter or the MTS_DISPATCHERS parameter's LISTENER attribute.

Note: The Oracle Database Configuration allows you to configure the LISTENER attribute. LOCAL_LISTENER must be configured manually.

LOCAL_LISTENER parameter configuration is shown below:

```
local_listener=local_listener_name
mts_dispatchers="(protocol=tcp | tcps)
(presentation=oracle.aurora.server.SGiopServer |
oracle.aurora.server.GiopServer)"
```

MTS_DISPATCHERS parameter's LISTENER attribute configuration is shown below:

```
mts_dispatchers="(protocol=tcp | tcps)
(presentation=oracle.aurora.server.SGiopServer |
oracle.aurora.server.GiopServer)(listener=local_listener_name)"
```

Parameter	Description
LISTENER (LIS or LIST)	Specifies the network name of the listener with which the dispatchers will register. It should be set to a name which is resolved through a naming method.

Remote Listener

If you want the connection to go through a *remote* listener, INITSID.ORA must be also configured with the MTS_DISPATCHERS parameter's LISTENER attribute:

```
mts_dispatchers="(protocol=tcp | tcps)
(presentation=oracle.aurora.server.SGiopServer |
oracle.aurora.server.GiopServer)(listener=remote_listener_name)"
```

Parameter	Description
LISTENER (LIS or LIST)	<p>The network name of the listener with which the dispatchers will register. It should be set to a name which is resolved through a naming method. This attribute should be only specified if the listener is at a remote location or is in another Oracle home.</p> <p>The LISTENER attribute makes it easier to administer multi-homed hosts.</p>

Step 2: Resolve the Listener Name

In order for the dispatcher to register its information with a non-default listener, the listener name you specified for the LOCAL_LISTENER parameter or the MTS_DISPATCHERS parameter's LISTENER attribute must be resolved through a naming method, such as a TNSNAMES.ORA on the server or an Oracle Names server.

The entry should only contain the address of the listener, not the service name information in the CONNECT_DATA portion.

Configuring a TNSNAMES.ORA File

The Net8 Assistant does not allow you to configure a TNSNAMES.ORA without the CONNECT_DATA information. Therefore, Oracle Corporation recommends you create the TNSNAMES.ORA manually. For example, if you configured a listener named LISTENER1 with TCP/IP on port 2481, the entry in the TNSNAMES.ORA would look similar to the entry below:

```
listener1.us.oracle.com=
(description=
  (address=
    (protocol= tcp)
    (host= sales-pc)
    (port= 2481)
  )
)
```

Multiple addresses are supported, but multiple address connect-time failover and client load balancing features are not supported. Oracle Connection Manager's SOURCE_ROUTE parameter is supported.

Configuring an Oracle Names Server

An entry for the listener can be made through an Oracle Names server through the Net8 Assistant:

1. Start Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Select the Oracle Names Servers folder.
3. Click on an Oracle Names server.
4. Select Manage Data from the drop-down list box.
5. Select the Advanced tab:
6. Click Add, and enter the listener name in the Name field, A.SMD record type in the Type field, and enter the address in the Value field, such as:

```
(address=(protocol= tcp)(host= sales-pc)(port= 2481))
```
7. Choose Save Network Configuration from the File menu.
8. Choose Exit from the File menu to exit the Net8 Assistant.

Configuration to Access a Dispatcher Directly

To configure IOP client connections to the Java option through a dispatcher, follow these steps:

- [Step 1: Enable MTS and Configure IOP in the INITSID.ORA File](#)
- [Step 2: Configure Clients](#)

Step 1: Enable MTS and Configure IOP in the INITSID.ORA File MTS must be configured in the `INITSID.ORA` file in the following manner:

```
mts_dispatchers="(address=(protocol=tcp | tcps)(host=server_
host)(port=port))(dispatchers=1)(presentation=oracle.aurora.server.SGiopServer |
oracle.aurora.server.GiopServer)"
```

The attributes are described below:

Parameter	Description
ADDRESS (ADD or ADDR)	Specifies the network address that the dispatchers will listen on. The network address may include either the TCP/IP (TCP) or the TCP/IP with SSL (TCPS) protocol, the host name of the server, and a GIOP listening port, which may be any port you choose that is not already in use.
DISPATCHERS (DIS or DISP)	The initial number of dispatchers to start.
PRESENTATION (PRE or PRES)	<p>Enables support for GIOP.</p> <p>Valid values for GIOP presentation values include:</p> <ul style="list-style-type: none"> ■ <code>oracle.aurora.server.SGiopServer</code> for session-based GIOP dispatcher connections. This presentation is valid for TCP/IP and TCP/IP with SSL. ■ <code>oracle.aurora.server.GiopServer</code> for standard GIOP dispatcher connections. This presentation is valid for TCP/IP and TCP/IP with SSL. <p>See the <i>Oracle8i Enterprise JavaBeans and CORBA Developer's Guide</i> for further information.</p>

Step 2: Configure Clients Configure the same port on the client as you used to configure MTS to access the dispatcher directly. See the *Oracle8i Enterprise JavaBeans and CORBA Developer's Guide* for further information.

Using SSL with EJBs and CORBA

Oracle8i also supports the use of authentication data such as certificates and private keys required for use by SSL in combination with both types of GIOP protocols - regular GIOP and session-based GIOP. To use SSL with GIOP, you need to carry out two steps:

1. Configure the listeners and dispatchers to accept SSL by specifying TCPS as the protocol.
 - If you want clients to access the database through the listener, follow the procedures in "[Non-Default Configuration for Access through a Listener](#)" on page 7-56.
 - If you want clients to access the database through a dispatcher directly, follow the procedures in "[Configuration to Access a Dispatcher Directly](#)" on page 7-63. Optionally, you can uncomment the following lines in the `INITSID.ORA` file that the Oracle Database Configuration Assistant creates for session-based IIOp and regular IIOp for TCP/IP with SSL:

```
#mts_dispatchers="(address=(protocol=tcp)(host=server_host)(port=5555))
(dispatchers=1)(presentation=oracle.aurora.server.SGiopServer)"
#mts_dispatchers="(address=(protocol=tcp)(host=server_host)(port=5556))
(dispatchers=1)(presentation=oracle.aurora.server.GiopServer)"
```

If you uncomment these lines, ensure the client is configured with same port number.

2. Specify the SSL wallet to be used by setting the `SQLNET.ORA` parameters as follows for both the listener and the dispatcher.
 - To configure the listener you specify:

```
ssl_client_authentication=false
ssl_version=undetermined
```
 - To configure the dispatcher, you specify:

```
oss.source.my_wallet=
  (source=
    (method=file)
    (method_data=
      (directory=wallet_location)))
```

See the *Oracle Advanced Security Administrator's Guide* for further information about configuring the SSL wallet.

Configuring Non-Oracle Database Services

Net8 supports connections to non-Oracle database services, including:

- [Configuring Net8 for External Procedures](#)
- [Configuring Net8 for Oracle Heterogeneous Services](#)
- [Configuring Net8 for an Oracle Rdb Database](#)

Configuring Net8 for External Procedures

The network listener may also be statically configured to listen for External Procedure Calls. When a PL/SQL or SQL application calls an external procedure, the Net8 network listener launches a network session-specific process called EXTPROC. Through the network listener service, PL/SQL passes the following information to EXTPROC.

- shared library name
- external procedure name
- parameters (if necessary)

EXTPROC then loads the shared library and invokes the external procedure.

External procedures are configured by default after installation.

For environments where the external procedure information does not exist, edit the LISTENER.ORA and TNSNAMES.ORA files, located in \$ORACLE_HOME/network/admin on UNIX and ORACLE_HOME\network\admin on Windows platforms, as follows:

1. Configure either a TCP/IP or IPC listener address. For example:

```
listener=
  (address=
    (protocol=ipc)
    (key=extproco0)
  )
```

For more information about configuring listener protocol addresses, see "[Configuring Listener Protocol Addresses](#)" on page 6-64.

2. Add a system identifier (SID) name of PLSEXPROC and a program name of EXTPROC in the server's LISTENER.ORA file (entries are shown in boldface text).

```

sid_list_listener_name
  (sid_list=
    (sid_desc=
      (sid_name=plsextproc)
      (oracle_home=d:\orant)
      (program=extproc)
    )
  )

```

Parameter	Description
SID_NAME	external procedure identifier
ORACLE_HOME	location of program executable
PROGRAM	external procedure program name

For more information about configuring listener protocol addresses, see ["Configuring a Listener with Service Information"](#) on page 6-71.

3. Add a net service name description entry for EXTPROC in the server's TNSNAMES.ORA file, using SID rather than SERVICE_NAME in the CONNECT_DATA section. For example:

```

extproc_connection_data=
  (description=
    (address=(protocol=ipc)(key=extproc0))

    (connect_data=
      (sid=plsextproc)
    )
  )

```

For more information about creating a net service name, see ["Configuring the Network with the Local Naming Method"](#) on page 6-12.

Configuring Net8 for Oracle Heterogeneous Services

Heterogeneous Services is an integrated component within the Oracle8i server, and provides the generic technology for accessing non-Oracle systems from the Oracle server. Heterogeneous Services enables you to:

- Use Oracle SQL to transparently access data stored in non-Oracle systems as if the data resides within an Oracle server.
- Use Oracle procedure calls to transparently access non-Oracle systems, services, or application programming interfaces (APIs), from your Oracle distributed environment.

While Heterogeneous Services provides the generic technology in the Oracle8i server, a Heterogeneous Services agent is required to access a particular non-Oracle system.

To initiate a connection to the non-Oracle system, the Oracle8i server starts an agent process through the Net8 listener on the gateway. For the Oracle8i server to be able to connect to the agent, perform the following steps:

1. On the Oracle8i database, set up a net service name to connect to the listener on the gateway. The connect descriptor must also include the HS=OK clause to make sure the connection uses Heterogeneous Services:
 - a. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
 - b. Create a net service name that will be used for connections from the Oracle8i server to a non-Oracle system.
See "[Step 1: Configure Net Service Names](#)" on page 6-14.
 - c. Double-click on the Net Service Names folder in the directory tree.
 - d. Select the net service name.
 - e. Click Advanced in the Service Identification group box.

The *Advanced Service Options* dialog box appears:



- f. Click *Use for Heterogeneous Service*, then click OK.
- g. Choose *Save Network Configuration* from the File menu.

The TNSNAMES.ORA file is re-created.

In the example below, a net service names in a TNSNAMES.ORA is configured for heterogeneous services:

```
megabase6_sales=
  (description=
    (address=
      (protocol=tcp)
      (host=dlsun206)
      (port=1521)
    )
    (connect_data=
      (service_name=sales6)
      (hs=ok)
    )
  )
```

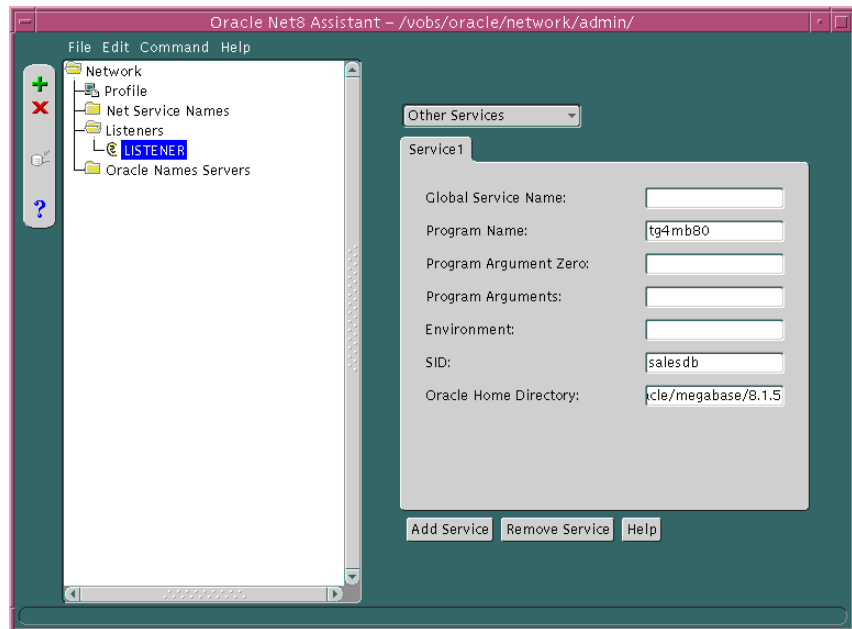
2. The listener on the gateway must be set up to listen for incoming requests from the Oracle8i server, and spawn heterogeneous services agents. The LISTENER.ORA file must be modified to set up the listener to start heterogeneous services agents, and the listener must be (re-)started. The features that must be set in the LISTENER.ORA include:

Net8 Assistant option	LISTENER.ORA Parameter	Description
Program Name	PROGRAM	Defines the name of the agent executable.
Oracle Home	ORACLE_HOME	The location of the agent executable. The agent executable must reside in the \$ORACLE_HOME/bin directory.
SID	SID_NAME	Specifies the Oracle System Identifier (SID) for the non-Oracle database.

To provide a listener with a new name use the Net8 Assistant:

- a. From the Net8 Assistant, double-click on the Listeners folder.
- b. Click on a listener.
- c. Choose Other Services from the drop-down list box.

- d. Click Add Service:



- e. Enter the program name that gets executed to create a gateway, the Oracle home for the executable, and the SID or service name of the non-Oracle system.
- f. Choose Save Network Configuration from the File menu.

The LISTENER.ORA file is re-created.

An excerpt of a LISTENER.ORA file configured for heterogeneous service in the SID_LIST_listener_name section is shown below:

```
sid_list_listener=(sid_list=
  (sid_desc=
    (sid_name=salesdb)
    (oracle_home=/home/oracle/megabase/8.1.5)
    (program=tg4mb80)
  )
)
```


- g. Choose Exit from the File menu.

The Net8 Assistant application exits.

For more information about Heterogeneous Services, see *Oracle8i Distributed Database Systems*.

Configuring Net8 for an Oracle Rdb Database

Oracle Rdb is a database for Digital's 64-bit platforms. With Net8, Oracle Rdb servers appears the same way to clients as Oracle8i databases. Because Oracle Rdb has its own listener, the client interacts with Rdb in the same manner as it does with Oracle7.

To initiate a connection to an Oracle Rdb, set up a net service name to connect to the Oracle Rdb database using the following features:

Net8 Assistant option	TNSNAMES.ORA Parameter	Description
RDB Database	RDB_DATABASE	Specifies the file name of an Oracle RDB database. Embed this parameter in the CONNECT_DATA section.
Type of Service	TYPE_OF_SERVICE	Specifies the type of service to use for an Oracle RDB database. It is used by Rdb interface tools. This feature should only be used if the application supports both an Oracle Rdb and Oracle database, and you want the application to randomly choose (load balance). Embed this parameter in the DESCRIPTION.
Global Database Name	GLOBAL_NAME	An optional feature, it identifies the Oracle Rdb database. Embed this parameter in the CONNECT_DATA section.

For further configuration information, see your Oracle Rdb documentation.

To configure a client for an Oracle Rdb database, use the Net8 Assistant:

- a. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
- b. Create a net service name that will be used for connections from the client to the Oracle Rdb.
- c. Double-click on the Net Service Names folder in the directory tree.
- d. Select the net service name.
- e. Click Advanced in the Service Identification group box.

"Step 1: Configure Net Service Names" on page 6-14.

The *Advanced Service Options* dialog box appears:



- f. Enter the file name of an *Oracle RDB* database in the Rdb Database field.
- g. Optionally, enter the global database name in the Global Database Name field, and, if needed, specify the type of service in the *Type of Service* field, then click OK.
- h. Choose Save Network Configuration from the File menu.

The `TNSNAMES.ORA` file is re-created.

In the example below, a net service name in a TNSNAMES.ORA is configured for an Oracle Rdb database:

```
alpha5=
  (description=
    (address= ...)
    (connect_data=
      (service_name=generic)
      (rbd_database= [.mf]mf_personnel.rdb)
      (global_name= alpha5)))
```

In the example below, TYPE_OF_SERVICE is used to load balance between an Oracle Rdb database service or Oracle8i database service:

```
alpha5=
  (description_list=
    (description=
      (address= ...)
      (connect_data=
        (service_name=generic)
        (rbd_database= [.mf]mf_personnel.rdb)
        (global_name= alpha5)))
    (description=
      (address= ...)
      (connect_data=
        (service_name=sales.com)
        (type_of_service=oracle8_database)))
```

Establishing a Connection and Testing the Network

Once you have completed configuring your network, you make a connection and test each component to ensure the network is functioning properly. Net8 provides a variety of tools to help you start, test and control a Oracle Names server, listener, and Connection Manager.

This chapter outlines procedures to make a connection and test network components using Net8's control utilities. This chapter contains the following sections:

- [Connecting to a Database](#)
- [Checklist for Troubleshooting Common Startup and Connection Problems](#)
- [Testing the Network](#)

Connecting to a Database

Connecting to a database involves starting network components and entering a connect string with a net service name, such as connect *username/password@net_service_name*.

This section covers the following topics:

- [Net8 Component Startup Overview](#)
- [Entering a Connect String](#)

Net8 Component Startup Overview

Client workstations and other servers connect to a listener with a net service name when logging onto an Oracle server.

After installing and configuring all the network components, you need to start them to make the network functional. Following is an outline of the steps you should take to start the network components.

1. If your network uses Oracle Names, start the Oracle Names servers, discover Oracle Names servers on all clients, and run the client cache.

Note: If the Oracle Names servers use a database to store the network information, you will need to start the database first, then start the listener.

2. Start the listeners using the Listener Control Utility, LSNRCTL.
3. Start the databases, using the tool of your choice.
4. If your network includes Oracle Connection Managers, start them using the Connection Manager Control Utility, CMCTL.

Step 1: Start Oracle Names Servers

Start Oracle Names servers using either Net8 Assistant or the NAMESCTL control utility on the machines where Oracle Names server software is installed and configured:

Use the control utility NAMESCTL...	Use the Net8 Assistant...
<p>From the command line, enter:</p> <pre>namesctl namesctl> start</pre> <p>The START command of NAMESCTL loads the Oracle Names server into memory and tells it to begin executing. At startup, the Oracle Names server loads its configuration and data.</p>	<ol style="list-style-type: none"> 1. Start Net8 Assistant. <ul style="list-style-type: none"> -On UNIX, run <code>netasst</code> at <code>\$ORACLE_HOME/bin</code>. -On Windows NT, choose Start > Programs > Oracle - <i>HOME_NAME</i> > Network Administration > Net8 Assistant. 2. Double-click on the Oracle Names server folder. 3. Select the Oracle Names server. 4. Select Manage Server from the drop-down list box. 5. Select the Control tab. 6. Click the Start radio button from the Server Operations field. 7. Click Apply.

On Windows NT, an Oracle Names server may also be started through the Control Panel:

1. Double-click the Services icon in the *Control Panel* window.
2. Select the Oracle *HOME_NAME* Names on *names_server* Service service.
3. Click Start to start the service.
4. In the *Services* window, click Close.

Step 2: Start Oracle Names Client Cache

To avoid clients looking up address information in an Oracle Names server each time, create a client cache of the information. This information is stored in CKPCCH.ORA located in `$ORACLE_HOME/network/names` on UNIX and `ORACLE_HOME\network\names` on Windows platforms.

To create a client cache:

1. Ensure discovery of Oracle Names servers has been performed, as described in ["Step 4: Configure Clients and Database Servers To Use Oracle Names Servers"](#) on page 6-37. The information gathered during discovery is used for a client's cache.
2. Create and start the client cache on the client:

```
namesctl  
namesctl> start_client_cache
```

On Windows NT, the client cache may also be started through the Control Panel:

1. Double-click the Services icon in the *Control Panel* window.
2. Select the OracleHOME_NAMEClientCache service.
3. Click Start to start the service.
4. In the *Services* window, click Close.

Step 3: Start the Listeners

For Net8 to accept connections on the server, a listener must be started with the LSNRCTL control utility on the server:

5. From the command line enter:

```
lsnrctl  
lsnrctl> status [listener_name]
```

where *listener_name* is the name of the listener defined in the LISTENER.ORA file. It is not necessary to identify the listener if you are using the default listener, named LISTENER.

If the STATUS command indicates the listener is running, go to Step 2. If the listener is not running, go to Step 3.

6. Even if the listener is running, Oracle Corporation advises you to stop the listener, and start it again. To stop the listener, enter:

```
lsnrctl> set password password  
lsnrctl> stop [listener_name]
```

SET PASSWORD is only required if the password is set in the LISTENER.ORA file. The password defaults to ORACLE.

7. Start the listener. Enter:

```
lsnrctl> start [listener_name]
```

LSNRCTL will display a status message indicating that the listener has started successfully. Check that all expected services for that listener are listed in the services summary in the status message.

8. Exit from the LSNRCTL utility. Enter:

```
lsnrctl> exit
```

On Windows NT, the listener may also be started through the Control Panel:

1. Double-click the Services icon in the *Control Panel* window.
2. Select the Oracle_{HOME_NAME}TNSListener service (the service name if you are using the default network listener name LISTENER) or Oracle_{HOME_NAME}TNSListener/*lsnr* (where *lsnr* is the non-default network listener name).
3. Click Start to start the service.
4. In the *Services* window, click Close.

Step 4: Start the Database

Use the tool of choice to start the database, such as SQL*Plus:

1. Start SQL*Plus without connecting to the database by entering:

```
sqlplus /nolog
```

2. Connect to Oracle as SYSDBA:

```
sql> connect username/password as sysdba
```

3. When you enter a STARTUP command, specify the database name and full path of the parameter file:

```
sql> startup database_name pfile=init $sid$ .ora
```

If you do not specify the PFILE option Oracle uses the standard parameter file location of `$ORACLE_BASE/admin/db_name/pfile/sid` on UNIX platforms and `ORACLE_BASE\admin\db_name\pfile\sid` on Windows NT; if you do not specify a database name Oracle uses the value for DB_NAME in the parameter file that starts the instance.

For further information on starting the database, see *Oracle8i Administrator's Guide*.

Step 5: Start Oracle Connection Managers

If Oracle Connection Manager is installed and configured, start it with the CMCTL control utility:

1. From the command line, enter:

```
cmctl  
cmctl> start cman
```

CMCTL displays a status message indicating that Oracle Connection Manager has started successfully.

2. Exit from the CMCTL utility. Enter:

```
cmctl> exit
```

On Windows NT, the listener may also be started through the Control Panel:

1. Double-click the Services icon in the *Control Panel* window.
2. If you are using Oracle Names, select the OracleHOME_NAMEECMAAdminService to acquire information about available Oracle Names Servers, then click Start. If you are not using Oracle Names, do not start this service.

The service starts.

3. Select the OracleHOME_NAMEECManService to start the Oracle Connection Manager, and click Start.

The service starts.

4. In the *Services* window, click Close.

Entering a Connect String

After the network components are started, as described in "[Connecting to a Database](#)" on page 8-2 you should be able to make a connection across the network. How you make a connection depends upon the naming method you configured in [Chapter 6, "Configuring Naming Methods and the Listener"](#) and the tool used for the connection. The basic connect strings to use are shown below:

For...	Enter...
Host Naming method	<pre>sql> CONNECT username/password@alias</pre> <p>The alias is typically set global database name in an IP address resolution mechanism. If the client and server are in the same domain, it is only necessary to enter the database name of the global database name.</p>
Local Naming method	<pre>sql> CONNECT username/password@net_service_name</pre>
Oracle Names method	<pre>sql> CONNECT username/password@net_service_name</pre>
Bequeath protocol	<pre>sql> CONNECT username/password</pre>

There are a number of ways to initiate a connection to an Oracle server. Commonly used methods include:

- [Connecting from the Operating System to Test a Client](#)
- [Connecting from the Tool Logon Screen to Test a Client](#)
- [Connecting from 3GL to Test a Client](#)
- [Connecting Using Special Commands within Tools](#)

The specifics of use are slightly different in each case. Each of the general methods listed is briefly covered here. To identify the method used in a specific tool, refer to the tool's user guide.

Connecting from the Operating System to Test a Client

The general form of connecting an application to a database server from the command line is:

```
tool username/password@net_service_name
```

```
sqlplus system/password@sales
```

To prevent the password from displaying during a logon, you can leave out the password parameter on the command line. For example:

```
sqlplus system@sales
```

You will be prompted to enter your password without it showing on screen.

Most Oracle tools can use the operating system command line to connect; some provide alternatives.

Connecting from the Tool Logon Screen to Test a Client

Some tools provide a logon screen as an alternative form of logon. A user can log on to a database server by identifying both the user name and net service name (*username@net_service_name*) in the user name field of the tool logon screen, and typing the password as usual in the password field.

Connecting from 3GL to Test a Client

In applications written using 3GL, the program must establish a connection to a server using the following syntax:

```
exec sql connect :username identified by :password
```

In this connection request, the *:username* and *:password* are 3GL variables that can be set within the program either statically or by prompting the user. When connecting to a database server, the value of the *:username* variable is in the form:

```
username@net_service_name
```

The *:password* variable contains the password for the database account being connected to.

Connecting Using Special Commands within Tools

Some Oracle tools have commands for database connection, once the tool has been started, to allow an alternative username to be specified without leaving the tool. SQL*Plus allow the CONNECT command using the following syntax:

```
sql> CONNECT username/password@net_service_name
```

For example:

```
sql> CONNECT scott/tiger@serverx
```

This is very similar to the operating system command line method, except that it is entered in response to the tool prompt instead of the operating system prompt.

Other Oracle tools use slightly different methods specific to their function or interface. For example, Oracle CDE tools use logon buttons and a pop-up window with the username, password, and remote database ID field. For more information on connecting to Oracle with a specific tool, refer to the tool's user guide.

Checklist for Troubleshooting Common Startup and Connection Problems

The following checklist is provided to help you troubleshoot common problems you may encounter when starting Net8 components or making a connection.

Problem	Work-Around
Inactive Components	<ol style="list-style-type: none">1. Verify that you have installed a transport layer protocol and a protocol.2. Verify that you have started a listener for any server you intend to contact.3. Start Oracle Connection Manager if you are routing sessions across protocols.
Syntax errors in your configuration files	<ol style="list-style-type: none">1. Use the Net8 Assistant whenever possible to avoid syntax errors. If you have manually created or edited configuration files, check them carefully to ensure that all the appropriate parentheses are in place, that the lines are indented to show their logical structure, and that there are no typographical errors. Additional Information: See "Syntax Rules for Configuration Files" on page C-2 for details on the syntax of these files.2. Verify that all net service names are mapped to connect descriptors in any applicable local naming configuration file.3. Verify that an invalid listener name was not typed in the LSNRCTL START command.4. Check your typing. Verify that the listener name you are using matches the name specified in your listener configuration file (LISTENER.ORA).

Problem	Work-Around
Files are incorrectly placed.	<ol style="list-style-type: none"> <li data-bbox="615 262 1273 661">1. The listener will indicate that it cannot start because configuration files could not be found. Normally, all configuration files are stored in <code>\$ORACLE_HOME/network/admin</code> on UNIX and <code>ORACLE_HOME/network/admin</code> on Windows NT. However, the environment variable or registry value, <code>TNS_ADMIN</code>, can be set to point to a different location. If <code>\$TNS_ADMIN</code> is set to a different directory, Net8 will expect your configuration files to exist in that directory. Secondly, Net8 looks in a <code>ORACLE_HOME</code> directory for configuration files. The <code>ORACLE_HOME</code> directory is different for each operating system. The best way to tell the location of the files is to look in your <code>SQLNET.LOG</code> file or in the header information in your trace file if you have asked for tracing to be turned on. <li data-bbox="615 678 1273 782">2. If you are using Oracle Names servers, verify that they have been started. Also make sure <code>SQLNET.ORA</code> contains a preferred Oracle Names server parameter so that a Names List file exists. <li data-bbox="615 800 1273 956">3. If using an external naming service such as NIS is in use, make sure that the appropriate external naming method has been installed on clients and servers, and that net service names have been properly loaded into it. Refer to your operating system-specific documentation for information.
The address is already in use.	<p>Another process may already be using the address listed in the listener configuration file (<code>LISTENER.ORA</code>). On some protocols such as TCP/IP, DECnet, and OSI, each network service on a node must use a unique port or socket. On other network protocols such as SPX or NetBIOS, each net service name must be unique for the entire network. Another network service may be using the same configuration. Contact your network administrator to evaluate whether the network address is available.</p>
<p>When trying to connect to a database, you may get the message <code>ORA-12203: "TNS:Unable to connect to destination"</code>.</p>	<p>Use the Listener Control Utility (<code>LSNRCTL</code>) to start the listener.</p>

Problem	Work-Around
<p>When trying to make a connection from a client, you may get the message ORA-12154: "TNS:Could not resolve service name".</p>	<ul style="list-style-type: none"> ■ The net service name you requested is not defined in your Oracle Names server, external naming method, or the local naming file can not be found as expected. ■ If you are using Oracle Names, this problem may indicate an Oracle Names server definition problem. Verify the SQLNET.ORA contains a NAMES.DIRECTORY_PATH parameter for a list of naming methods. ■ Server is not running ■ A Oracle Names server List file does not exist. Issue a REORDER_NS command from NAMESCTL. ■ The NAMES.PREFERRED_SERVER parameter is not configured correctly in a SQLNET.ORA file. Verify that the parameter is configured correctly in the SQLNET.ORA.
<p>When trying to connect to a database, you may get the message ORA-1034: "Oracle Not Available", or ORA-12505: "Listener could not resolve SID given in connect descriptor".</p>	<p>The database is not running on the server machine. A listener alone does not provide a database connection; the database instance must also be started.</p>
<p>A client returns the message "ORA-12541: No Listener".</p>	<p>Connect requests that come in too quickly for a listener to handle, and which exceed the listener's backlog (determined by QUEUESIZE parameter in LISTENER.ORA), are returned with an "Connection Refused" error. A client encountering this error returns the message "ORA-12541: No Listener" and the client log or trace files will show the "Connection Refused" message.</p> <p>To correct this problem, follow these steps:</p> <ol style="list-style-type: none"> 1. Stop the listener. 2. Reconfigure QUEUESIZE in your listener (LISTENER.ORA) configuration file to be a larger value (based on anticipated simultaneous connect requests). 3. Restart the listener. 4. Try to connect again.

Problem	Work-Around
When attempting to stop the listener, you may get the message TNS-01169: "The listener has not recognized the password."	Enter the SET PASSWORD command from within the Listener Control Utility (LSNRCTL), and then enter the STOP command to stop the listener.

Testing the Network

The preferred sequence for testing the network is as follows:

1. Start and test each Oracle Names server (if included in your network layout).
2. Start and test each listener.
3. Start and test each Oracle Connection Manager (if included in your network layout).
4. Test the server with a loopback test.
5. Test client with a connection.

This section cover the following topics:

- [Using Net8 Control Utilities](#)
- [Testing an Oracle Names Server](#)
- [Testing Network Objects Using NAMESCTL or the Net8 Assistant](#)
- [Testing a Listener](#)
- [Testing Oracle Connection Manager](#)
- [Testing Configuration on the Server](#)
- [Testing Network Connectivity from the Client](#)

Using Net8 Control Utilities

Net8 provides the following tools to help you start, test and control each network component.

- Oracle Names Control Utility — NAMESCTL
- Listener Control Utility — LSNRCTL
- Connection Manager Control Utility — CMCTL

For more information about Net8's component control utilities and their commands, refer to [Appendix A, "Control Utilities for the Listener, Oracle Names Server, and Oracle Connection Manager"](#).

Using the Oracle Names Control Utility (NAMESCTL)

The Oracle Names Control Utility, NAMESCTL, is a tool that you run from the operating system prompt to start and control the Oracle Names server.

The general form of the Oracle Names Control Utility is:

```
namesctl command
```

You can also issue NAMESCTL commands at the program prompt. When you enter NAMESCTL on the command line, the program is opened. You can then enter the desired commands from the program prompt. For example, the following command starts the Oracle Names server.

```
namesctl> start
```

Using the Listener Control Utility (LSNRCTL)

The Listener Control Utility, LSNRCTL, is a tool that you run from the operating system prompt to start and control the listener. The general form of the Listener Control Utility is:

```
lsnrctl command [listener_name] [args]
```

You can also issue Listener Control Utility commands at the program prompt. When you enter LSNRCTL on the command line, the program is opened. You can then enter the desired commands from the program prompt. For example, the following command determines the amount of time in seconds the listener will wait for a valid connection request after a connection has been started.

```
lsnrctl> set connect_timeout 20
```

Using the Connection Manager Control Utility (CMCTL)

The Connection Manager Control Utility, CMCTL, is a tool that you run from the operating system prompt to start and control Oracle Connection Manager. The general form of the Connection Manager Control Utility is:

```
cmctl command
```

You can also issue CMCTL commands at the program prompt. When you enter CMCTL on the command line, the program is opened. You can then enter the desired commands from the program prompt. For example, the following command starts Oracle Connection Manager.

```
cmctl> start
```

Testing an Oracle Names Server

To test an Oracle Names server, use the NAMESCTL PING command. Following are two ways to PING the Oracle Names server LABRADOR in the US.ACME domain.

From the NAMESCTL prompt, type:

```
namesctl> ping labrador.us.acme
```

You can test several Oracle Names servers with the same PING command. For example:

```
namesctl> ping huey.uk.acme duey.uk.acme louie.uk.acme
```

PING responds with the time it takes to contact the Oracle Names server and return an acknowledgment. If PING fails, make sure the Oracle Names server is started or double-check the configured address of the Oracle Names server.

Testing Network Objects Using NAMESCTL or the Net8 Assistant

The information stored in an Oracle Names server ("[Configuring the Network with the Oracle Names Method](#)" on page 6-21) can be queried to verify registration with the QUERY command.

To use QUERY command, use either the Net8 Assistant or the NAMESCTL control utility:

Use the Net8 Assistant...	Use the control utility NAMESCTL...
<ol style="list-style-type: none"> 1. Start the Net8 Assistant: <ul style="list-style-type: none"> -On UNIX, run <code>netasst</code> at <code>\$ORACLE_HOME/bin</code>. -On Windows NT, choose Start > Programs > Oracle - <code>HOME_NAME</code> > Network Administration > Net8 Assistant. 2. Double-click on the Oracle Names server folder. 3. Select an Oracle Names server. 4. Choose Manage Data from the drop-down list box. 5. Click the Advanced tab. 6. Select Query. 7. Enter the name of the object to query in the Name field. 8. Optionally, enter the type of the object in the Name field: <ul style="list-style-type: none"> -A.SMD: Network addresses, such a database network definitions for net service names. -CNAME.SMD: Alias name -DL.RDBMS.OMD: Global database link -DLCR.RDBMS.OMD: Link qualifier -NS.SMD: Oracle Names server address. System data used to communicate between Oracle Names servers -V1ADD.NPO.OMD: SQL*Net Version 1 connect string 9. Select Execute. 10. Select Save Network Configuration from the File menu. 11. Select Exit from the File menu to exit the Net8 Assistant application. 	<p>For a Oracle Names server in the administrative region, create and register an alias with the Oracle Names server:</p> <p>To display all data:</p> <pre>namesctl namesctl> query name *</pre> <p>The type of information to retrieve may also be requested. Common object types include:</p> <ul style="list-style-type: none"> ■ A.SMD: Network addresses, such a database network definitions for net service names. ■ CNAME.SMD: alias name ■ DL.RDBMS.OMD: global database link ■ DLCR.RDBMS.OMD: global link qualifier ■ NS.SMD: Oracle Names server address ■ V1ADD.NPO.OMD: SQL*Net Version 1 connect string <p>To display the database address for a net service name:</p> <pre>namesctl namesctl> query name a.smd</pre> <p>The following example shows a query of the net service name SALES:</p> <pre>namesctl namesctl> query sales a.smd</pre> <p>The QUERY command returns the amount of time the transaction took and information about the network object.</p>

Testing a Listener

To test a listener, initiate a connection from a client to any active database controlled by that listener, as described in "[Testing Configuration on the Server](#)" on page 8-18. If the only clients available to access the listener are on a different protocol, you must use an Oracle Connection Manager to access the listener.

Testing Oracle Connection Manager

To test Oracle Connection Manager, initiate a connection from a client to any active database for which a source route address has been created.

Testing Configuration on the Server

Once you have configured the network, test the configuration by performing a loopback test on the server.

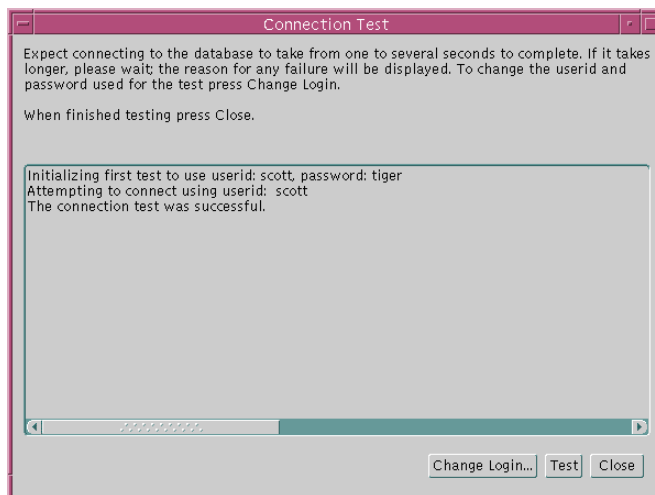
A loopback test uses Net8 to go from the server back to itself, bypassing the Interprocess Communication (IPC). Performing a successful loopback verifies that Net8 is functioning on the server side.

To perform the loopback test, use the Net8 Assistant:

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Click the Net Service Names folder in the directory tree.
3. Select the net service name.
4. Select Test Net Service Name from the Command menu.

Testing assumes the database and listener are running. If they are not, see "[Net8 Component Startup Overview](#)" on page 8-2 to start components.

A successful test results in "*The connect test was successful.*" message in the *Connect Test* dialog box:



If the test was not successful:

- Ensure the database and listener are running, then click Test.
 - Click Change Login to change the user name and password for the connection, then click Test.
5. Click Close to dismiss the *Connect Test* dialog box.

Testing Network Connectivity from the Client

To test several different clients in your network, initiate a connection to a server from each them by following the instructions in "[Entering a Connect String](#)" on page 8-8.

Net8 also provides the following tools to help you evaluate network connectivity:

- [TNSPING](#)
- [TRCROUTE](#)
- [Net8 Assistant](#)

TNSPING

TNSPING is a utility that determines whether or not a service (for example, an Oracle database, an Oracle Names server or any other Oracle service) on a Net8 network can be successfully reached.

If you can connect successfully from a client to a server (or a server to another server) using TNSPING, it displays an estimate of the round trip time (in milliseconds) it takes to reach the Net8 service.

If it fails, it displays a message describing the error that occurred. This allows you to see the network error that is occurring without the overhead of a database connection.

Using TNSPING To invoke the TNSPING utility, enter the following:

```
tnsping net_service_name [count]
```

Note: Different platforms may have different interfaces, but the program accepts the same arguments. Invoke TNSPING for the display of the proper interface requirements.

- *net service name*: must exist in TNSNAMES.ORA or the name service in use, such as NIS or DCE's CDS.
- *count* (optional): determines how many times the program attempts to reach the server.

If the net service name specified is a database name, TNSPING attempts to contact the corresponding listener. It does not actually determine whether or not the database itself is running. Use Server Manager to attempt a connection to the database.

Following are some examples of TNSPING.

Reaching a Database To connect to a database using a net service name of SPOTDB, the following is entered:

```
tnsping spotdb
```

This produces the following message:

```
TNS Ping Utility for SunOS:
Copyright (c) Oracle Corporation 1998. All rights reserved.
Attempting to contact
  (ADDRESS=(PROTOCOL=TCP)(HOST=spot)(PORT=1521))
OK (50msec)
```

To determine whether the STPRD database can be connected to, and to specify that TNSPING try to connect 10 times and then give up, use the following command:

```
tnsping stprd 10
```

This command produces the following message:

```
TNS Ping Utility for SunOS:
Copyright (c) Oracle Corporation 1998. All rights reserved.
Attempting to contact (ADDRESS=(PROTOCOL=TCP)(HOST=spot)(PORT=1521))
OK (290 msec)
OK (100 msec)
OK (70 msec)
OK (70 msec)
OK (60 msec)
OK (70 msec)
OK (70 msec)
OK (80 msec)
OK (180 msec)
OK (340 msec)
```

Invalid Net Service Name Below is an example of TNSPING attempting to connect to an invalid net service name:

```
tnsping bad_db
```

This attempt produces the following message:

```
TNS Ping Utility for SunOS:  
Copyright (c) Oracle Corporation 1998. All rights reserved.  
TNS-03505: Failed to resolve name
```

Valid Net Service Name Without Listener Following is an example of using TNSPING to connect to a name that is valid, but that resolves to an address where no listener is located (for example, the listener may not be started):

```
tnsping testing
```

The following message is returned:

```
TNS Ping Utility for SunOS:  
Copyright (c) Oracle Corporation 1998. All rights reserved.  
Attempting to contact (ADDRESS=(PROTOCOL=tcp)(HOST=spot)(PORT=1521))  
TNS-12541: TNS:no listener
```

Reaching an Oracle Names Server To check whether a Oracle Names server can be reached, use a command using the Net8 address as in the following:

```
tnsping (address=(protocol=tcp)(host=fido)(port=1575))
```

A message similar to the following will be returned to the user:

```
TNS Ping Utility for SunOS:  
Copyright (c) Oracle Corporation 1998. All rights reserved.  
Attempting to contact (ADDRESS=(PROTOCOL=TCP)(HOST=fido)(PORT=1575))  
OK (70 msec)
```

TRCROUTE

The Trace Route Utility (TRCROUTE) enables administrators to discover what path or route a connection is taking from a client to a server. If TRCROUTE encounters a problem, it returns an error stack to the client instead of a single error. These additional error messages make troubleshooting easier.

TRCROUTE is different from TNSPING in that it travels as a special type of connect packet, and is routed as such. As it travels toward its destination, the TRCROUTE connect packet collects the TNS addresses of every node it travels through. If an error occurs, TRCROUTE collects error information that shows where the error occurred. The Trace Route Utility displays the information collected on the client screen. You can redirect the TRCROUTE output to a file, and print it if you wish.

Requirements Trace Route works only over Net8 and SQL*Net version 2.3 and later. Every node along the route from client to server must use SQL*Net version 2.3 or later. If a pre-2.3 node is on the path, the following error is displayed:

```
TNS-03603: Encountered a node with pre-2.3 version of SQL*Net
```

TRCROUTE shows what node along the path is responsible for any errors.

Effect on Performance The Trace Route Utility uses minimal resources. It gathers information in the connect data of a special connect packet; standard connect packets are not affected.

The server is not affected by TRCROUTE. The listener receives and processes the TRCROUTE connect packet. It returns the information to the client by putting it into a refuse packet. The server does not need to start up any new processes or deal with dummy connections.

Using TCROUTE To invoke TRCROUTE, type the following from the command line:

```
trcroute net_service_name
```

If you have configured your network to use listener load balancing, there may be more than one listener on different nodes for a database. If so, the Trace Route Utility might use any of the listeners, just as a regular connection request might. The output it returns shows you what listener node it used.

The following are two examples of trace route output:

- a successful Trace Route packet that traveled from a client to a listener
- an unsuccessful Trace Route packet that could not reach the listener because the listener was not up.

Figure 8–1 Successful Trace Route

```
%trcroute tcp_direct
Trace Route Utility for Solaris:
Copyright (c) Oracle Corporation 1998. All rights reserved.

Route of TRCROUTE:-----
Node: Client           Time and address of entry into node:
-----
01-DEC-96 13:26:36 ADDRESS= PROTOCOL=TCP Host=shining-sun Port=1581
Node: Server           Time and address of entry into node:
-----
01-DEC-96 13:27:20 ADDRESS= PROTOCOL=TCP Host=setting-sun Port=1521
```

Figure 8–2 Trace Route with Error

```
% trcroute tcp_direct

Trace Route Utility for SVR4:
Copyright (c) Oracle Corporation 1996. All rights reserved.

Route of TRCROUTE:-----
Node: Client           Time and address of entry into node:
-----
01-DEC-96 11:12:34 ADDRESS= PROTOCOL=TCP Host=shining-sun Port=1581
TNS-12224: TNS:no listener
TNS-12541: TNS:no listener
TNS-12560: TNS:protocol adapter error
TNS-03601: Failed in route information collection
```

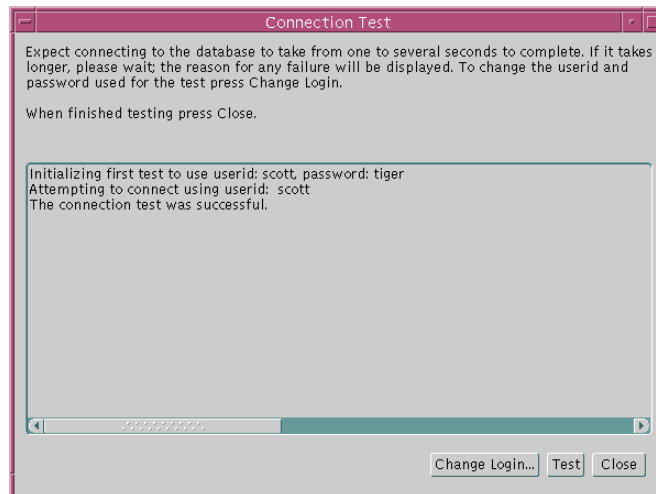
Net8 Assistant

To verify connectivity for a client machine, use the Net8 Assistant:

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Click the Net Service Names folder in the directory tree.
3. Select the net service name.
4. Select Test Net Service Name from the Command menu.

Testing assumes the database and listener are running. If they are not, see "[Net8 Component Startup Overview](#)" on page 8-2 to start components.

A successful test results in "*The connect test was successful.*" message in the *Connect Test* dialog box:



If the test was not successful:

- Ensure the database and listener are running, then click Test.
 - Click Change Login to change the user name and password for the connection, then click Test.
5. Click Close to dismiss the *Connect Test* dialog box.

Configuring Multi-Threaded Server

This chapter describes how to configure multi-threaded server (MTS).

Specific topics discussed are:

- [Overview](#)
- [Configuring Oracle for Multi-Threaded Server Architecture](#)
- [Configuring Both Multi-Threaded Server and Dedicated Server Modes](#)
- [Overriding Multi-Threaded Server Mode on the Client](#)

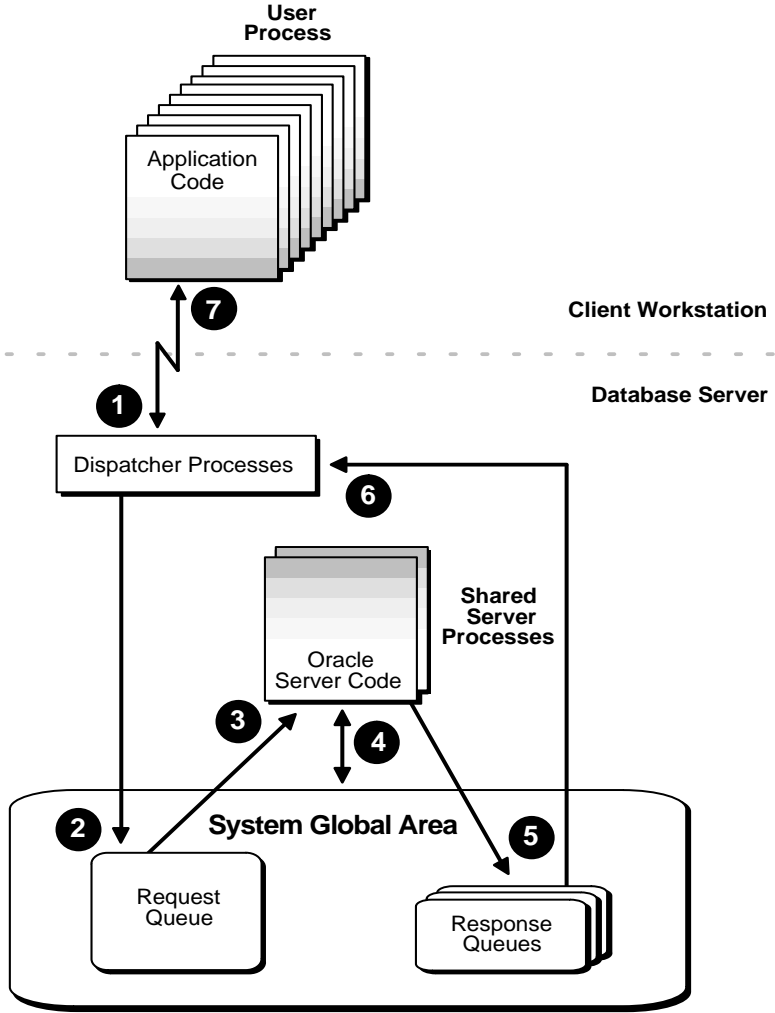
Overview

Consider an order entry system with a dedicated server architecture. A customer places an order as a clerk enters the order into the database. For most of the transaction, the clerk is on the telephone talking to the customer and the server process dedicated to the clerk's user process remains idle. The server process is not needed during most of the transaction, and the system is slower for other clerks entering orders because the idle server process is holding system resources.

The MTS architecture eliminates the need for a dedicated server process for each connection (see [Figure 9-1](#)). A small number of shared servers can perform the same amount of processing as many dedicated servers. Also, since the amount of memory required for each user is relatively small, less memory and process management are required, and more users can be supported.

Features such as connection pooling help increase user scalability by increasing the number of possible simultaneous connections to the database. See "[Enabling Connection Concentration](#)" on page 7-31 and "[Connection Pooling](#)" on page 2-50.

Figure 9-1 Oracle Multi-Threaded Server Architecture



Configuring Oracle for Multi-Threaded Server Architecture

To set up your system in a MTS configuration, start the listener process and set the `MTS_DISPATCHERS` parameter in the initialization file (`INITSID.ORA`).

Note: The Oracle Database Configuration Assistant, available on Windows NT and some UNIX platforms, allows you to configure this parameter.

After setting this parameter, restart the instance, which at this point will use the MTS configuration. `MTS_DISPATCHERS` should be set in the following manner:

```
mts_dispatchers="(attribute=value)"
```

One and only one of the following attributes is required: `ADDRESS`, `DESCRIPTION`, or `PROTOCOL`.

Attribute	Description
<code>ADDRESS</code> (<code>ADD</code> or <code>ADDR</code>)	The network protocol address of the end point which the dispatchers will listen. See " Configuring Protocol Addresses " on page 6-76 for further information about protocol address syntax.
<code>DESCRIPTION</code> (<code>DES</code> or <code>DESC</code>)	The network description of the end point which the dispatchers will listen on, including the protocol. The syntax is as follows: <code>(description =(address=...))</code>
<code>PROTOCOL</code> (<code>PRO</code> or <code>PROT</code>)	Specifies the network protocol for which the dispatcher will generate a listening end point.

The attributes CONNECTIONS, DISPATCHERS, LISTENER, MULTIPLEX, POOL, PRESENTATION, SERVICE, and TICKS are optional:

Attribute	Description
CONNECTIONS (CON or CONN)	<p>The maximum number of network connections to allow for each dispatcher.</p> <p>The default is operating-system specific.</p>
DISPATCHERS (DIS or DISP)	<p>The initial number of dispatchers to start. The default is 1.</p>
LISTENER (LIS or LIST)	<p>The network name of the listener with which the dispatchers will register. It should be set to a name which is resolved through a naming method. This attribute should be only specified if the listener is:</p> <ul style="list-style-type: none"> ■ a non-default local listener ■ at a remote location <p>A non-default local listener may also be specified with the LOCAL_LISTENER parameter. Oracle Corporation recommends using the LISTENER attribute, as the Oracle Database Configuration Assistant does not support configuration of the LOCAL_LISTENER parameter.</p> <p>Important: In order for the dispatcher to register its information with a non-default listener, the listener name specified must be resolved through a naming method, such as a TNSNAMES.ORA on the server or an Oracle Names server. The entry should only contain the address of the listener, not the service name information in the CONNECT_DATA portion.</p> <p>For example, if the listener name is LIST1, the listening endpoint uses TCP/IP on port 1521, and the chosen naming method is the local naming method, the entry in the TNSNAMES.ORA would be:</p> <pre>listener1.us.oracle.com= (description= (address= (protocol= tcp) (host= sales2-pc) (port= 1521)))</pre> <p>The Net8 Assistant does not allow you to configure a TNSNAMES.ORA without the CONNECT_DATA information. Therefore, Oracle Corporation recommends you create the TNSNAMES.ORA manually.</p>

Attribute	Description
MULTIPLEX (MUL or MULT)	<p>Used to enable Oracle Connection Manager's connection concentration feature.</p> <p>If 1, ON, YES, TRUE, or BOTH is specified, then connection concentration is enabled for both incoming and outgoing network connections.</p> <p>If IN is specified, the connection concentration is enabled for incoming network connections from the client.</p> <p>If OUT is specified, the connection concentration is enabled for outgoing network connections from the client.</p> <p>If 0, NO, OFF, or FALSE is specified, then connection concentration is disabled for both incoming and outgoing network connections.</p> <p>Additional Information: See "Enabling Connection Concentration" on page 7-31.</p>
POOL	<p>If a number is specified, then connection pooling is enabled for both incoming and outgoing network connections and the number specified is the timeout in ticks for both incoming and outgoing network connections.</p> <p>If ON, YES, TRUE, or BOTH is specified, then connection concentration is enabled for both incoming and outgoing network connections and the default timeout of 10 will be used for both incoming and outgoing network connections.</p> <p>Additional Information: See "Enabling Connection Pooling" on page 7-49 and "Connection Pooling" on page 2-50.</p>

Attribute	Description
PRESENTATION (PRE or PRES)	<p>Enables support of specific presentation protocols, including:</p> <ul style="list-style-type: none"> ■ Two-Task Common (TTC) for standard Net8 and Java Stored Procedures connections (Default) ■ General Inter-Orb Protocol (GIOP) for Enterprise Java Beans and CORBA Server connections that use the Internet Inter-Orb Protocol (IIOP) <p>IIOP is an implementation of GIOP over TCP/IP or TCP/IP with SSL. Valid values for GIOP presentation values include:</p> <ul style="list-style-type: none"> ■ oracle.aurora.server.SGiopServer for session-based GIOP dispatcher connections. This presentation is valid for TCP/IP and TCP/IP with SSL. ■ oracle.aurora.server.GiopServer for standard GIOP dispatcher connections. This presentation is valid for TCP/IP and TCP/IP with SSL. <p>Each presentation requires a separate MTS_DISPATCHERS entry.</p> <p>Additional Information: See "Configuring Java Option Connections" on page 7-50.</p>
SERVICE (SER, SERV)	<p>The service name which the dispatchers register with the Net8 listeners. The SERVICE attribute overrides the SERVICE_NAMES parameter.</p>
SESSIONS (SES or SESS)	<p>The maximum number of network sessions to allow for each dispatcher.</p> <p>The default is operating-system specific.</p>
TICKS	<p>The size of a network tick in seconds. The default is 15 seconds. This parameter is optional for connection pooling.</p>

See *Oracle8i Tuning* for further information on tuning MTS parameters.

Setting the Initial Number of Dispatchers

The number of dispatchers started at instance startup is controlled by the parameter `MTS_DISPATCHERS`. Estimate the number of dispatchers to start for each network protocol before instance startup.

Note: Unlike the shared servers the number of dispatchers does not change dynamically. The number of dispatchers must be explicitly changed with the `ALTER SYSTEM` command. You can change the number of number of dispatchers in this manner up to a maximum limit of `MTS_MAX_DISPATCHERS`.

`MTS_MAX_DISPATCHERS` has a default value of 5. If the initial number of dispatchers, the sum of all dispatchers configured in multiple `MTS_DISPATCHERS` variable, configured at setup time is greater than the value specified for `MTS_MAX_DISPATCHERS`, then the system automatically resets `MTS_MAX_DISPATCHERS` to the larger number.

The appropriate number of dispatchers for each instance depends upon the performance you want from your database, the host operating system's limit on the number of connections per process, (which is operating system dependent) and the number of connections required per network protocol.

The instance must be able to provide as many connections as there are concurrent users on the database system. After instance startup, you can start more dispatchers if needed.

Calculating the Initial Number of Dispatcher

Once you know the number of possible connections per process for your operating system, calculate the initial number of dispatchers to create during instance startup, per network protocol, using the following formula.

$$\begin{array}{l} \text{number} \\ \text{of} \\ \text{dispatchers} \end{array} = \text{CEIL} \left(\frac{\text{maximum number of concurrent sessions}}{\text{connections per dispatcher}} \right)$$

For example, assume that your system typically has 900 users concurrently connected via TCP/IP and 600 users connected via SPX, and supports 255 connections per process. In this case, the DISPATCHERS attribute for TCP/IP should be set to a minimum of 4 dispatchers and SPX should be set to a minimum of 3 dispatchers:

```
mts_dispatchers="(protocol=tcp) (dispatchers=4) "
mts_dispatchers="(protocol=spx) (dispatchers=3) "
```

Examples

Example 1 To force the IP address used for the dispatchers, enter the following:

```
mts_dispatchers="(address=(partial=true)(protocol=tcp)(host=144.25.16.201))
(dispatchers=2) "
```

This will start 2 dispatchers that will listen in on HOST=144.25.16.201, which must be a card that is accessible to the dispatchers.

Example 2 To force the exact location of dispatchers, add the PORT as follows:

```
mts_dispatchers="(address=(partial=true)(protocol=tcp)
(host=144.25.16.201)(port=5000))(dispatchers=1) "
mts_dispatchers="(address=(partial=true)(protocol=tcp)
(host=144.25.16.201)(port=5001))(dispatchers=1) "
```

Note: You can specify multiple MTS_DISPATCHERS in the INITSID.ORA file, but they must be adjacent to each other. Also, MTS_DISPATCHERS defaults to 1.

Configuring Both Multi-Threaded Server and Dedicated Server Modes

Oracle8i release 8.1.5 supports MTS for GIOP and TTC presentations. (IIOP is an implementation of GIOP over TCP/IP.) Because each presentation requires its own MTS_DISPATCHERS entry in the INITSID.ORA to enable MTS connections, it is possible to configure MTS for IIOP clients and dedicated server for TTC clients. This is achieved by setting the MTS_DISPATCHERS parameter in the INITSID.ORA file with a GIOP presentation. Because dedicated server is the default behavior, no further configuration for TTC is required. A dedicated server will be used for all TTC client connections.

In the example below, two MTS_DISPATCHERS entries are configured for different GIOP presentation protocols, indicated by the PRESENTATION attribute:

```
mts_dispatchers="(protocol=tcp)
(presentation=oracle.aurora.server.SGiopServer) "
mts_dispatchers="(protocol=tcp)
(presentation=oracle.aurora.server.GiopServer) "
```

It is also possible to configure MTS for IIOP clients and TTC clients. This is achieved by setting separate entries for each presentation. In the example below, three MTS_DISPATCHERS entries are configured for:

- TTC for Net8 clients
Because TTC is the default presentations, there is no explicit entry for it
- two GIOP presentation protocols for IIOP clients

```
mts_dispatchers="(protocol=tcp) "
mts_dispatchers="(protocol=tcp)
(presentation=oracle.aurora.server.SGiopServer) "
mts_dispatchers="(protocol=tcp)
(presentation=oracle.aurora.server.GiopServer) "
```


Overriding Multi-Threaded Server Mode on the Client

If the network is configured for MTS and a particular client requires dedicated server mode, the TNSNAMES.ORA file can be configured with a net service name that uses a dedicated server. This is achieved with (SERVER=DEDICATED) in the CONNECT_DATA section of a connect descriptor:

```
net_service_name=  
(description=  
  (address= (protocol_address_information))  
  (connect_data=  
    (service_name=service_name)  
    (server=dedicated)  
  )  
)
```

See "[Configuring Advanced Connection Attributes for a Net Service Name](#)" on page 7-7 to set the SERVER parameter.

Enabling Net8 Enhancements for Programmers

Net8 includes an application program interface (API) called Net8 OPEN allowing programmers to develop both database and non-database applications. In addition, Net8 contains several new benefits for programmers, including UNIX client programming, signal handler and alarm programming, Bequeath protocol, and child process termination.

This chapter contains the following sections:

- [Net8 OPEN](#)
- [UNIX Client Programming](#)

Net8 OPEN

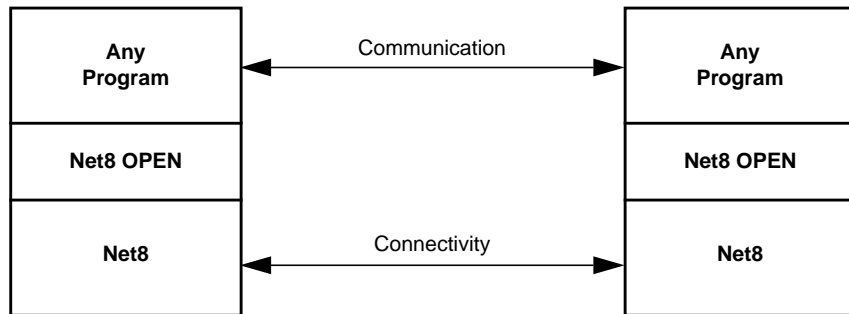
Net8 includes an application program interface (API) called Net8 OPEN, which enables programmers to:

- develop both database and non-database applications that make use of the Net8 network already deployed in their environment
- deploy an application developed on one machine to another without having to modify their calls to the network interface

Net8 OPEN provides applications a single common interface to all industry standard network protocols.

The relationship of Net8 OPEN to other products is shown in [Figure 10–1](#).

Figure 10–1 Net8 OPEN



Using Net8 OPEN, you can solve a number of problems, such as:

- Three-tier connectivity (client/agent/server) - Use any application to communicate with an agent. For example, the agent might be an application server that allows simultaneous connectivity to Oracle and non-Oracle data sources, such as remote information servers.
- Distributed applications - Build distributed applications that can run over an existing Oracle network without the requirements of a database or additional middleware.
- Enhanced clients - Integrate non-SQL information with SQL applications. For example, a process control application can communicate with a non-SQL application such as a sensor.

Net8 OPEN API Function Calls

In contrast to a remote procedure call interface, Net8 OPEN provides a byte stream-oriented API that can be used to develop basic applications which send and receive data. Applications developed with Net8 OPEN must ensure that values sent across the network are interpreted correctly at the receiving end.

The Net8 OPEN API consists of five function calls:

- [TNSOPEN](#)
- [TNSCLOSE](#)
- [TNSSEND](#)
- [TNSRECV](#)
- [TNSCONTROL](#)

TNSOPEN

Description:	<p>Initializes the Net8 OPEN API per-connection handle. This function must be the first Net8 OPEN call that a user makes. Note that <code>tnsopen()</code> does not establish a connection; connections are established by the send and receive calls as needed.</p> <p>If you are writing a client program (which will initiate the connection), "name" contains a net service name in the same format as those in a <code>TNSNAMES.ORA</code> file.</p> <p>If you are writing a server program, "name" should be <code>NULL</code>. Your server program will pick up the connection automatically when you make the first <code>tnsrecv()</code> call to receive data (see the section on <code>tnsrecv</code>).</p>
Synopsis:	<pre>int tnsopen(handlep, name) void **handlep; const char *name;</pre>
Parameters:	<p><code>handlep</code> (IN/OUT) - Address to receive Net8 connection handle</p> <p><code>name</code> (IN) - net service name</p>
Prerequisites:	The <code>handlep</code> parameter must not be <code>NULL</code>
Returns:	Upon successful completion a zero value is returned. Otherwise, a positive Net8 API error is returned.

TNSCLOSE

Description:	Shuts down the connection. The user must call this function to close the connection and release the handle properly.
Synopsis:	<pre>int tnsfclose(handlep) void **handlep;</pre>
Parameters:	handlep (IN/OUT) - Address of a pointer to a Net8 connection handle
Prerequisites:	The handlep parameter must not be NULL.
Returns:	Upon successful completion a zero value is returned, and *handlep is set to NULL. Otherwise, a positive Net8 API error number is returned.

TNSSEND

Description:	<p>Sends data to the Net8 connection handle.</p> <p>In the first call to <code>tnssend()</code> on the client side, the connection is established before any data is sent to the handle. The client application must first call <code>tnsopen()</code> to establish a connection to the server. It is an error if the client application calls <code>tnsrecv()</code> first, or if the server program calls <code>tnssend()</code> first.</p> <p>Note that this also means that the <code>tnssend()</code> call may return errors related to connection establishment - so the first indication you get that, for instance, you have given the incorrect TNS address, is that an error occurs on the first <code>tnssend()</code> call, rather than on the <code>tnsopen()</code> call as you may first expect.</p>
Synopsis:	<pre>int tnssend(handle, data, length) void *handle; const void *data; size_t *length;</pre>
Parameters:	<p><code>handle</code> (IN/OUT) - pointer to Net8 connection handle returned by <code>tnsopen()</code></p> <p><code>data</code> (IN) - pointer to data to be sent</p> <p><code>length</code> (IN/OUT) - pointer to the length of data to be sent in bytes and the actual number of bytes written on return.</p>
Prerequisites:	The parameters must not be NULL.
Returns:	Upon successful completion a zero value is returned, and the actual number of bytes written is returned as the value pointed to by the <code>length</code> parameter. Otherwise, a positive Net8 API error number is returned.

TNSRECV

Description:	<p>Receives data from the Net8 connection handle.</p> <p>In the first call to <code>tnsrecv()</code> on the server side, the connection is established before data is received from the handle. The server must first call <code>tnsrecv()</code> after <code>tnsopen()</code> to accept the connection from the client.</p> <p>Incoming connections are accepted by the Net8 Listener (<code>tnslsnr</code>), which automatically spawns off a copy of your server program when needed, and hands it the new connection. You must configure the network listener with the location of your server program for this to occur — see the section on configuration below.</p>
Synopsis:	<pre>int tnsrecv(handle, data, length) void *handle; void *data; size_t *length;</pre>
Parameters:	<p><code>handle</code> (IN/OUT) - pointer to Net8 connection handle returned by <code>tnsopen()</code></p> <p><code>data</code> (IN/OUT) - pointer to buffer to receive data</p> <p><code>length</code> (IN/OUT) - pointer to the length of buffer to receive data and actual number of bytes received on return</p>
Prerequisites:	All parameters must not be NULL.
Returns:	Upon successful completion a zero value is returned, and the actual number of bytes received is returned as the value pointed to by the length parameter. Otherwise, a positive Net8 API error number is returned.

TNSCONTROL

Description:	Sets the connection to blocking or nonblocking mode.
Synopsis:	<pre>int tnscontrol(handle, cmd) void *handle; int *cmd;</pre>
Parameters:	<p><code>handle (IN)</code> - pointer to Net8 connection handle returned by <code>tnsopen()</code></p> <p><code>cmd (IN)</code> - option to apply to the connection. Currently supported values are:</p> <p><code>TNSAPINONBLOCKING</code> - set connection into nonblocking mode</p> <p><code>TNSAPIBLOCKING</code> - set connection into blocking mode</p>
Prerequisites:	The handle must not be NULL, and cmd must be one of the supported commands.
Returns:	A zero value is returned if the option is successfully set.

Finding the Net8 OPEN Applications Program Interface

The applications program interface is provided as part of the standard Net8 installation. To use it, you need the following:

- **TNSAPI.H** - this is the header file which describes the API interfaces and errors. It is provided in `$ORACLE_HOME/network/public` on UNIX and `ORACLE_HOME\network\tnsap\include` on Windows NT.
- The **Net8 OPEN library** - located with other Oracle networking libraries, and contains the name "TNSAPI". Note that the name of the library varies by platform. On UNIX, it is in your `$ORACLE_HOME/network/lib` directory and is named `LIBTNSAPI.A`. On Windows platforms, the `ORACLE_HOME\network\tnsap\lib` contain the files `TNSAPI.DLL` and `TNSAPI.LIB`.
- **Sample makefiles** - are provided for your platform in your network directory. They can be used to determine the appropriate link line to build your application.

Building Your Own Application

Modules which make reference to Net8 OPEN functions should include `TNSAPI.H`, as follows:

```
#include <tnsap.h>
```

Your makefile (or other location for your build command) should ensure that the include path is set properly so that it can find `TNSAPI.H`. Refer to the sample makefiles provided in your installation.

Configuring the System to Use Your Net8 OPEN Application

To configure Net8 to recognize your Net8 OPEN application, proceed as follows:

1. Add the location of your server program to your listener configuration file (LISTENER.ORA), so that the network listener knows to start your server if a connection request is received.

Note: This is not necessary for release 8.1 configurations, as database instance registration registers service information with the listener.

To do this, choose a system identifier (SID) name for your service similar to that of an Oracle database. Do not pick the same SID as your database.

For example, if you are configuring a "chat" program, you could call the SID "chatsid". Place the program into the same place as the Oracle server executable, which is normally `$ORACLE_HOME/bin` on UNIX and `ORACLE_HOME\bin` on Windows NT.

You would place the following entry in a listener configuration file as follows:

```
sid_list_listener =
  (sid_list =
    (sid_desc =
      (sid_name = chatsid)/*your SID name*/
      (oracle_home = /usr/oracle)/*$ORACLE_HOME bin directory*/
      (program = chatsvr)/*the name of your server program*/))
```

You need to restart the listener, so it will recognize the new service.

2. Add the address of your application server to your local names configuration file (TNSNAMES.ORA).

For example, if your listener is listening on the following address:

```
(description=(address=(protocol=tcp)(host=unixhost)(port=1521)))
```

And you want people to refer to the service you created above as "chat".

You would add the following parameter to your local naming configuration file for release 8.1 configuration:

```
chat=  
(description=  
  (address=(protocol=tcp)(host=unixhost)(port=1521)  
  )  
  (connect_data=(service_name=chatsid)))
```

You would add the following parameter to your local naming configuration file for pre-release 8.1 configuration:

```
chat=  
(description=  
  (address=(protocol=tcp)(host=unixhost)(port=1521)  
  )  
  (connect_data=(sid=chatsid)))
```

Note that the address contains the SID you configured in the LISTENER.ORA file above. Also note that the second line started with at least one space character, which indicates that it is a continuation line.

If you have domains in your network, you need to name your service accordingly. For instance, use chat.acme.com if the domain is acme.com. Again, use the TNSNAMES.ORA file as a template — if all the other net service names end in a domain, you need to name your service similarly.

3. Place the executable for your service in the same directory as your Oracle Server executable. On UNIX platforms, place the executable in the \$ORACLE_HOME/bin directory indicated in your LISTENER.ORA file. In this example, you would place the program "chatsvr" in the location /usr/oracle/bin/chatsvr.

If needed on your operating system, you also must ensure that you have permission to execute your program.

Sample Programs

Two sample applications are provided with Net8 OPEN:

- **finger** - this is a utility that connects to the server that retrieves information about who is logged in. This utility includes a pair of programs which demonstrate the basic steps involved in building a distributed application. The client program runs on both Solaris and Windows NT; the server is UNIX specific.
- **tftp** - this sample client and server program is implemented in UNIX to help you with simple file transfers using the tftp protocol.

Net8 OPEN API Errors

This section lists the error numbers which can be returned if one of the above function calls fails. Note that in some cases, connection-related errors may come back from a send or receive call, if the connection has not yet been established at that time.

- 20002 - SDFAIL_TNSAPIE - The underlying "send" command failed in tnsend().
- 20003 - RECVFAIL_TNSAPIE - The underlying "receive" command failed in tnsrecv().
- 20004 - INVSROP_TNSAPIE - Operation is invalid as the server.
- 20005 - INVCLIOP_TNSAPIE - Operation is invalid as the client.
- 20006 - HDLUNINI_TNSAPIE - The connection should be initialized by calling tnsopen().
- 20007 - INHFAIL_TNSAPIE - Server failed in inheriting the connection from the listener.
- 20008 - ACPTFAIL_TNSAPIE - Server failed in accepting the connection request from the client.
- 20009 - NULHDL_TNSAPIE - A null handle was passed into the call, which is not allowed.
- 20010 - INVOP_TNSAPIE - An invalid operation called was passed into the call.
- 20011 - MALFAIL_TNSAPIE - A malloc failed in TNS API call.
- 20012 - NLINIFAIL_TNSAPIE - Failed in NL initialization.
- 20013 - NMTOOLONG_TNSAPIE - Service name is too long.
- 20014 - CONFFAIL_TNSAPIE - Client connect request failed.
- 20015 - LSNFAIL_TNSAPIE - Server failed to listen for connect request.
- 20016 - ANSFAIL_TNSAPIE - Server failed to answer connect request.
- 20017 - NMRESFAIL_TNSAPIE - Failed to resolve service name.
- 20018 - WOULDBLOCK_TNSAPIE - Operation would block.
- 20019 - CTLFAIL_TNSAPIE - Control call failed.
- 20020 - TNSAPIE_ERROR - TNS error occurred.
- 20021 - INVCTL_TNSAPIE - Invalid operation request in control call.

UNIX Client Programming

Event programming in UNIX requires the use of a UNIX signal. When an event occurs, a signal flags a process. The process executes code that is relevant to the particular signal generated. UNIX does not allow a single process to set more than one signal handler or alarm for a particular signal call. If a process sets a second signal handler or alarm request on a signal like SIGCHLD (signal on a child process' status change), UNIX nullifies and loses the previous request for the SIGCHLD.

If any part of your application issues one of these requests, signal handling or alarms may cause the system to lose and never respond to that particular request. Depending on the signal requested, the system may not clean up defunct processes properly because of a signal handler problem.

Net8 provides two solutions to allow for the use of signal handling and alarms in tandem with Oracle's usage of those requests:

- [Signal Handler and Alarm Programming](#)
- [Bequeath](#)

Signal Handler and Alarm Programming

Net8 provides an operating system dependent (OSD) call that keeps a table of all signal handler or alarm requests for each signal. Any program that uses the signal handler or alarm is now required to use the Oracle OSD calls. This provides a solution for programmers in UNIX who are not allowed to set more than one signal handler or alarm for a particular call. Any program that uses the signal handler or alarm must use the Oracle OSD calls. This is however, currently available only for internal use. In the near future, an externalized version of the OSD calls for client application usage will be released.

Until then, if you set all of the client's signal handlers before making any database connections, the OSD call will remember the last signal handler set for the signal and will add it to the signal handler table. Note that by doing this, you cannot disable the signal handler.

Oracle OSD Signal Handling Rules

To use the table-driven shared OSD signal handler for all SIGCHLD calls, you must observe the following rules:

- Know your child process IDs so you can clean up the correct process.
- Use the `waitpid()` call instead of `wait()` on the correct child process ID.
- The `waitpid()` call must be non-blocking.

Bequeath

This section is for UNIX application programmers who use both the UNIX signal handler for tracking child process status changes with the SIGCHLD call and Net8 for the networking portion of their application.

When a client application is directed to communicate with an Oracle database on the same machine, it uses the Bequeath protocol to establish the connection. The Bequeath protocol enables the client to retrieve information from the database without using the listener. The Bequeath protocol internally spawns a server process for each client application. In a sense, it performs locally the same operation that a remote listener does for your connection.

Child Process Termination

Since the client application spawns a server process internally through the Bequeath protocol as a child process, the client application becomes responsible for cleaning up the child process when it completes. When the server process completes its connection responsibilities, it becomes a defunct process. Signal handlers are responsible for cleaning up these defunct processes. Alternatively, you may configure your client `SQLNET.ORA` file to pass this process to the UNIX init process by disabling signal handlers.

Use the Net8 Assistant to configure a client to disable the UNIX signal handler. The `SQLNET.ORA` parameter set to disable is as follows:

```
bequeath_detach=yes
```

This parameter causes all child processes to be passed over to the UNIX init process (pid = 1). The init process automatically checks for "defunct" child processes and terminates them.

Bequeath automatically chooses to use a signal handler in tracking child process status changes. If your application does not use any signal handling, then this default does not affect you.

Troubleshooting Net8

Net8 provides methods for understanding and resolving network problems through the use of log and trace files. These files keep track of the interaction between network components as errors occur. Evaluating this information will help you to diagnose and troubleshoot even the most complex network problems.

This chapter describes common network errors and outlines procedures for resolving them. It also describes methods for logging and tracing error information to diagnose and troubleshoot more complex network problems. This chapter contains the following sections:

- [Diagnosing Net8](#)
- [Resolving the Most Common Error Messages](#)
- [Troubleshooting Tips from the Field](#)
- [Troubleshooting Network Problems Using Log and Trace Files](#)
- [Logging Error Information](#)
- [Tracing Error Information](#)
- [Contacting Oracle Customer Support](#)

Diagnosing Net8

If you have just completed installing and configuring Net8 and an attempt to make a basic peer-to-peer (single protocol network) connection returns an ORA ERROR, this section may help you diagnose the cause of the problem.

Any underlying fault, noticeable or not, is reported by Net8 with an error number or message that is not always indicative of the actual problem. This section helps you determine which parts of Net8 do function properly rather than the parts that do not work. It also helps you to decide in which of the following categories the fault belongs:

- Oracle software
- Operating system layer
- Other network layers

Testing the various network layers progressively should in most cases uncover any problem.

Server Diagnostics

Answer the questions below: (You may need assistance from your server administrator to follow the instructions in this section.)

- Is any other system (workstations/servers) able to connect to the server using Net8?
- Has the server, database, or listener configuration remained the same for some time?

If you answered YES to any of the above questions/statements, skip this section and continue to "[Client Diagnostics](#)" on page 11-4.

If you are unsure, or answered NO to any of the above questions, please continue.

Diagnosing Net8 on the server involves the following tasks:

- [Task 1: Verify the Database Is Running](#)
- [Task 2: Perform a Loopback Test](#)

Task 1: Verify the Database Is Running

To check that the database is up:

- Log onto the database and connect with a valid user name and password. For example:

```
sqlplus system/manager
```

A message appears, confirming that you are connected with the database. If you receive the following errors, ask your Database Administrator to assist you:

- ORA-1017: invalid U/P
- ORA-1034: Oracle not available

Task 2: Perform a Loopback Test

To perform a loopback test:

1. Check that the LISTENER.ORA, TNSNAMES.ORA, and SQLNET.ORA files exist in `$ORACLE_HOME/network/admin` on UNIX and `ORACLE_HOME\network\admin` on Windows NT.
2. Follow the instructions in "[Testing Configuration on the Server](#)" on page 8-18 to perform a loopback test.
 - If the loopback test continues to fail, continue to the next step.
 - If the loopback test passes, skip to "[Client Diagnostics](#)" below.
3. Check the **Problem/Solution Database** website at <http://support.oracle.com> for more specific information on the error received, or contact Oracle Worldwide Support.

Client Diagnostics

At this point, you know the Net8 server side listener works properly, because you could verify at least one of the following statements:

- The server passed a loopback test, showing that the connection worked
- Other machines (servers, workstation) connect also using Net8 to this same Oracle Server.
- Connections from this workstation worked previous to making changes on this machine (such as the installation of a new product or a modification to the network configuration).

To perform diagnostics on the client:

1. Check that you have installed the same Oracle protocol(s) as were installed on the server. On UNIX, you can run the `adapters` program to verify this. Run `adapters` at `$ORACLE_HOME/bin`.

Output similar to the following appears:

Installed Net8 Transport Protocols are:

```
IPC
TCP/IP
BEQueath
SSL
RAW
...
```

2. Check base connectivity for underlying network transport. (You may need help from your network administrator to verify this.) Net8 technology depends on the underlying network for a successful connection.

Protocol	Verify that you can...
TCP/IP	Use file transfer or terminal emulation utilities (FTP, TELNET, and PING) from the workstation to the server where the listener and database reside.
SPX	<ul style="list-style-type: none"> ■ Perform a Netware log in to the machine on which the database is running. ■ Ensure you can map drives or use other Novell services such as Print Servers and File Servers on the Network. ■ Check that the listener service is broadcasting by doing a DISPLAY SERVERS from the Novell Server or any Novell File Server on the SPX network.
Named Pipes	<ul style="list-style-type: none"> ■ See other computers or servers on the MSFT network. ■ Ensure you are able to share drives within the MSFT network.

3. Verify that all Net8 software has been installed to ensure that both the Net8 Client and the appropriate protocol are present.
4. Ensure that the client machine has the TNSNAMES.ORA and the SQLNET.ORA files in `$ORACLE_HOME/network/admin` on UNIX and `ORACLE_HOME\network\admin` on Windows platforms.

The search order for SQLNET.ORA and TNSNAMES.ORA follows:

- a. Current working directory from where an application is executed
- b. TNS_ADMIN environment variable

If the TNS_ADMIN environment variable is not defined on Windows platforms, TNS_ADMIN it may be in the registry
- c. If TNS_ADMIN is not defined, `$ORACLE_HOME/network/admin` on UNIX and `ORACLE_HOME\network\admin` on Windows platforms

If you have any other working client machines connecting to your selected Oracle database using Net8, back up your existing files and copy both the working TNSNAMES.ORA and SQLNET.ORA files from the working machine onto the non-working client workstations. This eliminates the possibility of errors in the files.

5. Test the Net8 layer, as described in "[Testing Network Connectivity from the Client](#)" on page 8-20.

Note: Do *not* to use TNSPING. TNSPING works just like the TCP/IP PING utility and will *not* create and open a socket, nor does it connect with the listener. It just ensures listener is present at the server side.

6. If the connection still fails:
 - Use tracing as described in the following section "[Troubleshooting Network Problems Using Log and Trace Files](#)" on page 11-15.
 - Check the **Problem/Solution Database** website at <http://support.oracle.com> for a specific diagnostics bulletin on the error received.
 - Contact Oracle Worldwide Support.

Resolving the Most Common Error Messages

Due to the complexity of network communications, network errors may originate from a variety of sources, for a variety of reasons. If an error occurs, applications such as SQL*Plus and SQL*Forms, which depend on network services from Net8, will normally generate an error message.

A list of the most common network error messages follows:

- `ORA-12154: TNS:could not resolve service name`
- `ORA-12198: TNS:could not find path to destination`
- `ORA-12203 TNS:unable to connect to destination`
- `ORA-12224:TNS:no listener`
- `ORA-12500: TNS:listener failed to start a dedicated server process`
- `ORA-12533: TNS:illegal ADDRESS parameters`
- `ORA-12545: TNS:name lookup failure`
- `ORA-12560: TNS:protocol adapter error`
- `ORA-3113: TNS:End of file on communication channel`
- `ORA-3121: No interface driver connection - function not performed`

The table below describes each network error and outlines procedures to troubleshoot them.

Error #: Message	Description/Troubleshooting Procedures
ORA-12154: TNS:could not resolve service name	<p>Cause: Net8 could not locate the net service name specified in the TNSNAMES.ORA configuration file.</p> <p>Actions:</p> <ol style="list-style-type: none"> 1. Verify that a TNSNAMES.ORA file exists and that it is accessible. 2. Verify that there are not multiple copies of the TNSNAMES.ORA file. 3. In your TNSNAMES.ORA file, verify that the net service name specified in your connect string is mapped to a connect descriptor in the TNSNAMES.ORA file. Also, verify that there are no syntax errors in the file. 4. Verify that there are no duplicate copies of the SQLNET.ORA file. 5. If you are using domain names, verify that your SQLNET.ORA file contains a NAMES.DEFAULT_DOMAIN parameter. If this parameter does not exist, you must specify the domain name in your connect string. If you are not using domain names, and this parameter exists, delete it or disable it by commenting it out. 6. If you are connecting from a login dialog box, verify that you are not placing an "@" symbol before your connect net service name. 7. Activate client tracing and re-execute the operation.
ORA-12198: TNS:could not find path to destination	<p>Cause: The client is not able to find the desired database.</p> <p>Actions:</p>
ORA-12203 TNS:unable to connect to destination	<ol style="list-style-type: none"> 1. Verify that you have entered the net service name you wish to reach correctly. 2. Verify that the net service name ADDRESS parameters in the connect descriptor of your TNSNAMES.ORA file are correct. 3. Verify that your TNSNAMES.ORA file is stored in the correct directory. 4. Verify that the listener on the remote node has started and is running. If not, start the listener by using the Listener Control Utility. 5. If you are connecting from a login box, verify that you are not placing an "@" symbol before your connect net service name.

Error #: Message	Description/Troubleshooting Procedures
ORA-12203 TNS:unable to connect to destination	<p>ORA-12203 error is a generic error that often shields secondary errors. For this reason, check the latest SQLNET.LOG file located in \$ORACLE_HOME/network/log on UNIX and ORACLE_HOME\network\log on Windows platforms for secondary ORA messages. If after analyzing the log file you determine there are no secondary errors, determine if the problem may be caused by one the following scenarios:</p> <p>Cause: The incorrect Oracle protocol for the selected networking protocol is installed. A missing protocol support driver usually produces the following errors in the SQLNET.LOG or any client trace file:</p> <ul style="list-style-type: none"> ■ ORA-12203 ■ ORA-12538 ■ ORA-00508 <p>Action: Check that you have installed the appropriate Oracle protocol. On UNIX, you can run the adapters program to verify this. Run adapters at \$ORACLE_HOME/bin.</p> <p>Output similar to the following appears:</p> <p>Installed Net8 Transport Protocols are:</p> <pre> IPC TCP/IP BEQueath SSL RAW ... </pre>
ORA-12203 continued	<p>Cause: An invalid net service name was supplied in the connect string.</p> <p>Action: Verify that the net service name supplied in your connect string exists in your TNSNAMES.ORA file and the ADDRESS information for that net service name is valid. Ask yourself the following questions:</p> <ul style="list-style-type: none"> ■ Is the HOST or SERVICE name correct? ■ Is the PORT specified correct?

Error #: Message	Description/Troubleshooting Procedures
ORA-12203 continued	<p>Cause: Net8 could not find the connect descriptor specified in the TNSNAMES.ORA file.</p> <p>Action: After verifying that the database is running, check the following:</p> <ol style="list-style-type: none">1. Verify the listener is running. Enter: <pre>lsnrctl lsnrctl> status listener_name</pre>where <i>listener_name</i> is the name of the listener defined in the LISTENER.ORA file. It is not necessary to identify the listener if you are using the default listener, named LISTENER. If the output indicates the listener is not running, try starting it with the command: <pre>lsnrctl> start listener_name</pre>2. Ensure the TNSNAMES.ORA file is in the correct location, \$ORACLE_HOME/network/admin on UNIX and ORACLE_HOME\network\admin on Windows platforms.
ORA-12203 continued	<p>Cause: The destination system's listener is not listening.</p> <p>Action: Verify that the remote system's listener is running. Enter: <pre>lsnrctl lsnrctl> status listener_name</pre>where <i>listener_name</i> is the name of the listener defined in the LISTENER.ORA file. It is not necessary to identify the listener if you are using the default listener, named LISTENER. If the output indicates the listener is not running, try starting it with the command: <pre>lsnrctl> start listener_name</pre></p>
ORA-12203 continued	<p>Cause: <i>There are underlying network transport problems.</i></p> <p>Action: Verify with utilities supplied with the networking protocol being used that the protocol itself is functional. For example, with TCP/IP, try to PING the remote system.</p>
ORA-12203 continued	<p>Cause: TNSNAMES.ORA file is not located in the proper directory.</p> <p>Action: Make sure the TNSNAMES.ORA file is located in \$ORACLE_HOME/network/admin on UNIX and ORACLE_HOME\network\admin on Windows platforms or an alternative path, as explained in "Client Diagnostics" on page 11-4.</p>

Error #: Message	Description/Troubleshooting Procedures
ORA-12203 continued	<p>Cause: The (HOST=<i>server_name</i>) parameter for TCP/IP addresses or (SERVICE=<i>tns_application</i>) parameters for SPX addresses is not consistent on the clients and server machines.</p> <p>Action: Ensure the values for these parameter are the same on the server and client.</p> <p>For TCP/IP setups, make sure that the HOST parameter in LISTENER.ORA on the server and in the TNSNAMES.ORA file on the client point to the same name, or at least to names that are then translated to the same IP address by each system. This is especially important for servers with multiple IP addresses assigned to the various network interfaces on the server.</p> <p>For SPX setups, the name must be the same on the server and client workstations.</p>
ORA-12224:TNS:no listener	<p>Cause: The connection request could not be completed because the listener is not running.</p> <p>Actions:</p> <ol style="list-style-type: none"> 1. Ensure that the supplied destination address matches one of the addresses used by the listener. 2. Verify also that this is not a version compatibility problem.
ORA-12500: TNS:listener failed to start a dedicated server process	<p>Cause: The listener was unable to start a process connecting the user to the database server.</p> <p>Actions:</p> <ol style="list-style-type: none"> 1. Verify that the SID_LIST section of the LISTENER.ORA file and the system identifier (SID) in the CONNECT DATA section of the TNSNAMES.ORA file are correct. 2. Check that the program name specified by the PROGRAM parameter in the SID_LIST section exists in the /bin directory or specify the path for the program name. 3. Verify that the user has adequate privileges to access the database.
ORA-12533: TNS:illegal ADDRESS parameters	<p>Cause: The protocol specific parameters in the ADDRESS section of the designated connect descriptor in your TNSNAMES.ORA file are incorrect.</p> <p>Action: For more information about protocol specific keywords, refer to the Oracle operating system specific documentation for your platform.</p>

Error #: Message	Description/Troubleshooting Procedures
ORA-12545: TNS:name lookup failure	<p>Cause: The listener on the remote node cannot be contacted.</p> <p>Actions:</p> <ol style="list-style-type: none">1. Verify that the ADDRESS in the TNSNAMES.ORA file or the LISTENER.ORA file is correct.2. Verify that the listener on the remote node has been started. You may check its status with the STATUS command of the Listener Control Utility, and start it with the START command if necessary.
ORA-12560: TNS:protocol adapter error	<p>Cause: The listener was unable to start a process connecting the user to the database server.</p> <p>Actions:</p> <ol style="list-style-type: none">1. Turn on tracing and re-execute the operation.2. Evaluate the contents of the trace file to diagnose the problem.
ORA-3113: TNS:End of file on communication channel	<p>Cause: An unexpected end of file was processed on the communication channel. This may be an indication that the communications link may have gone down at least temporarily; it may indicate that the server has gone down.</p> <p>Action: You may need to modify your re-transmission count. For more information about troubleshooting this error, refer to the appropriate Oracle operating system specific documentation.</p>
ORA-3121: No interface driver connection - function not performed	<p>Cause: A SQL*Net version 1 prefix was erroneously used in the connect string.</p> <p>Action: Do not use the following prefixes in the connect string.</p> <ul style="list-style-type: none">■ T:■ X:■ P: <p>Cause: Only the user name and password were specified from a client machine that had no local Oracle database installed.</p> <p>Action: Specify a connect string.</p>

Troubleshooting Tips from the Field

Below are some tips you may find helpful when you are having difficulty diagnosing network problems:

- **Use the node or network address during configuration instead of the name of the server machine**—This eliminates any internal lookup problems and make the connection slightly faster.

TCP/IP—Use the internet address rather than the host name in TCP/IP addresses, for example, 198.32.3.5. Change the (HOST=*server_name*) line in the TNSNAMES.ORA file with the internet address, for example (HOST=198.32.3.5).

- **Consider possible SPX connection issues**—The workstation requesting a connection be made with a remote listener must first learn the location of that SPX service in the NetWare IPX network.

The client workstation issues a lookup request for the SPX service. If the service cannot be found, an error is sent back to the workstation.

- **Perform a loopback test**—Perform a loopback test on the server, as described in "[Testing Configuration on the Server](#)" on page 8-18. If the test passes, FTP the TNSNAMES.ORA and SQLNET.ORA files to the client.
- **Check what is between you and the server**—If it is a wide area network (WAN), identify any intermediate systems that may not work correctly. If all machines are fine, the problem may be a timing issue.
- **Verify Whether or Not There Is a Timing Issue**—Timing issues are associated with ORA-12203, ORA-12535, or ORA-12547 errors in the client log files.

To resolve this, try speeding up the connection by using exact addresses instead of names and increase the CONNECT_TIMEOUT parameter in the LISTENER.ORA file. The default value for this parameter is 10 seconds.

- **Determine which Oracle applications are failing**—SQL*Plus may work, but CASE tools may not. If you determine the problem is a data volume issue, try to transfer a large (5 MB) file with the base connectivity.

Questions to Ask When Troubleshooting

Below are some questions to ask yourself when diagnosing a problem:

- *Do all machines have a problem, or is it just one?*

If one machine works and another does not, and you are confident that the same software (Oracle and third-party products) is installed, swap out the network cables, if they are close enough, to see if the problem moves. If it does move, it indicates that the problem has something to do with the client-server connection and is not local to the PC.

- *What kind of links exist between the client and the server, for example, X.25, ISDN, Token Ring, or leased line?*

Sniffers and LAN analyzers are useful for intermittent failing connections or detecting time-outs and resent packets. You can also see what side of the conversation is waiting for a response.

- *Does the third-party application fail, but Oracle applications work?*

Troubleshooting Network Problems Using Log and Trace Files

Net8 provide detailed information about the source and context of problems as they arise. This information is generated and stored in log and trace files. The process of logging and tracing error information will help you to diagnose and resolve network problems.

For server and listener, log files are by default located in `$ORACLE_HOME/network/log` on UNIX and `ORACLE_HOME\network\log` on Windows platforms, and trace files are by default located in `$ORACLE_HOME/network/trace` on UNIX and `ORACLE_HOME\network\trace` on Windows platforms. For client, log and trace files are by default located in the current working directory.

Logging Error Information

All errors encountered in Oracle network products are appended to a log file for evaluation by a network or database administrator. The log file provides additional information for an administrator when the error message on the screen is inadequate to understand the failure. The log file, by way of the error stack, shows the state of the software at various layers.

To ensure that all errors are recorded, logging cannot be disabled on clients or Names Servers. Furthermore, only an administrator may replace or erase log files. The log file for the listener also includes Audit Trail information about every client connection request, as well as most listener control commands.

This section covers the following topics:

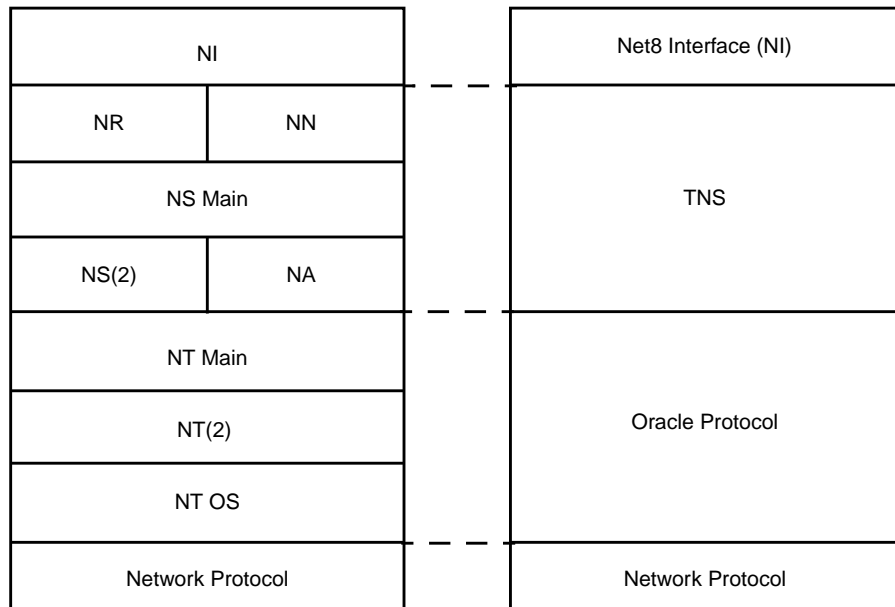
- [Error Stacks](#)
- [Log File Names](#)
- [Understanding and Setting Log Parameters](#)
- [Setting Log Settings During Runtime of Control Utilities](#)
- [Using Log Files](#)
- [Listener's Log Audit Trail](#)
- [Understanding Oracle Connection Manager Logs](#)

Error Stacks

Log files provide information contained in an error stack. An error stack refers to the information that is produced by each layer in an Oracle communications stack as the result of a network error.

Figure 11-1 depicts the relationship among Oracle network products as they might appear in an error stack:

Figure 11-1 Network Products and Error Stack Component



The layers in [Figure 11-1](#) are as follows:

NI	Net8 Interface Layer
NR	Network Routing
NN	Network Naming (Oracle Names)
NS	Network Session (main and secondary layers)
NA	Native Services includes Network Authentication (NA) and Network Encryption (NAE)
NT	Network Transport (main, secondary, and operating system layers)

Your network may or may not include all of these components.

Error Example

As an example, suppose that a user of a client application tries to establish a connection with a database server using Net8 and TCP/IP, and the user enters:

```
sqlplus scott/tiger@hrserver.com
```

The SQL*Plus banner is displayed on the screen, and the following error is displayed:

```
ORA-12203: TNS:Unable to connect to destination
```

This message indicates that the connection to the server failed because the database could not be contacted. Although the application displays only a one-line error message, an error stack that is much more informative is recorded in the log file by the network layer. On the client-side, a log file called SQLNET.LOG, contains an error stack corresponding to the ORA-12203 error as follows:

Figure 11-2 Typical Error Stack

```
*****
```

```
Fatal OSN connect error 12203, connecting to:
  (DESCRIPTION=(CONNECT_DATA=(SID=trace)(CID=(PROGRAM=)
    (HOST=lala)(USER=sviavant)))(ADDRESS_LIST=(ADDRESS=
    (PROTOCOL=ipc)(KEY=trace))(ADDRESS=(PROTOCOL=tcp)
    (HOST=lala)(PORT=1521))))
```

```
VERSION INFORMATION:
```

```
TNS for SunOS:
Oracle Bequeath NT Protocol Adapter for SunOS:
Unix Domain Socket IPC NT Protocol Adaptor for SunOS:
TCP/IP NT Protocol Adapter for SunOS:
  Tracing to file: /home/sviavant/trace_admin.trc
  Tns error struct:
    nr err code: 12203
    TNS-12203: TNS:unable to connect to destination
    ns main err code: 12541
    TNS-12541: TNS:no listener
    ns secondary err code: 12560
    nt main err code: 511
    TNS-00511: No listener
    nt secondary err code: 61
    nt OS err code: 0
```

Log File Names

Each Net8 component produces its own log file. The table below provides the default file names and a description of the information they contain:

Log File	Contains Error Information about the...
SQLNET.LOG	Client and/or server
LISTENER.LOG	Listener
NAMES.LOG	Oracle Names Server
CMAN_PID.TRC	Oracle Connection Manager CMGW gateway process
CMADM_PID.TRC	Oracle Connection Manager CMADMIN administrative process

Understanding and Setting Log Parameters

Parameters that control logging, including the type and amount of information logged, as well as the location where the files are stored, are set in the configuration file of each network component as follows:

These log parameters corresponding to the...	...are set in the following Configuration Files
client	SQLNET.ORA
server	SQLNET.ORA
listener	LISTENER.ORA
Oracle Names Server	NAMES.ORA
Oracle Connection Manager processes	CMAN.ORA

This section cover the following topics:

- [SQLNET.ORA Parameters](#)
- [LISTENER.ORA Parameters](#)
- [NAMES.ORA Parameters](#)
- [CMAN.ORA Parameters](#)
- [Setting Log Parameters in Configuration Files](#)

For further information about the parameters described in this section, see [Appendix C, "Configuration Parameters"](#).

SQLNET.ORA Parameters

The following parameters settings can be set in SQLNET.ORA:

SQLNET.ORA Parameter	Net8 Assistant Option	Description
LOG_DIRECTORY_CLIENT	Client Information: Log Directory	Establishes the destination directory for the client log file. By default, the client directory is the current working directory.
LOG_FILE_CLIENT	Client Information: Log File	Sets the name of the log file for the client. By default the log name is SQLNET.LOG.
LOG_DIRECTORY_SERVER	Server Information: Log Directory	Establishes the destination directory for the server log files. By default the server directory is <code>\$ORACLE_HOME/network/log</code> on UNIX and <code>ORACLE_HOME\network\log</code> on Windows NT.
LOG_FILE_SERVER	n/a	Sets the name of the log file for the server. By default the log name is SQLNET.LOG.

LISTENER.ORA Parameters

The following log parameters can be set in LISTENER.ORA:

LISTENER.ORA Parameter	Net8 Assistant Option	Description
LOG_DIRECTORY_ <i>listener_name</i>	Log File	Establishes the destination directory and file for the log file that is automatically generated for listener events. By default the directory is \$ORACLE_HOME/network/log on UNIX and <i>ORACLE_HOME</i> \network\log on Windows NT, and the file name is LISTENER.LOG.
LOG_FILE_ <i>listener_name</i>		

NAMES.ORA Parameters

The following log parameters can be set in NAMES.ORA:

NAMES.ORA Parameter	Net8 Assistant Option	Description
NAMES.LOG_DIRECTORY	Log Directory	Establishes the destination directory for log files. By default, the directory is \$ORACLE_HOME/network/log on UNIX and <i>ORACLE_HOME</i> \network\log on Windows NT.
NAMES.LOG_FILE	Log File	Sets the name of the log file for the client. By default the log name is NAMES.LOG.

CMAN.ORA Parameters

The following log parameter can be set in CMAN.ORA:

CMAN.ORA Parameter	Description
LOG_LEVEL	<p>Establishes the level of logging:</p> <ul style="list-style-type: none">range is 0 to 4:<ul style="list-style-type: none">level 0 - no logginglevel 1 - basic reportinglevel 2 - RULE_LIST matching lookup reportinglevel 3 - relay blocking reportinglevel 4 - relay I/O counts reporting <p>The CMGW gateway process creates a log file called <code>CMAN_PID.LOG</code>, and the CMADMIN administrative process creates a log file called <code>CMADM_PID.LOG</code>. The log files are located in <code>\$ORACLE_HOME/network/log</code> on UNIX and <code>ORACLE_HOME\network\log</code> on Windows NT.</p>

Setting Log Parameters in Configuration Files

SQLNET.ORA, LISTENER.ORA and NAMES.ORA logging parameters can be set with the Net8 Assistant. CMAN.ORA logging parameter must be set manually in CMAN.ORA, as described in "[Oracle Connection Manager Parameters \(CMAN.ORA\)](#)" on page C-76. To set logging parameters:

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Specify the log parameters:

For this log file...	Set logging parameters here...
SQLNET.LOG	<ol style="list-style-type: none"> 1. Click the Profile icon. 2. Choose General from the drop-down list box. 3. Click the Logging tab. 4. Specify the settings.
LISTENER.LOG	<ol style="list-style-type: none"> 1. Click the Listeners folder. 2. Select a listener. 3. Choose General Parameters from the drop-down list box. 4. Click the Logging and Tracing tab. 5. Specify the settings.
NAMES.LOG	<ol style="list-style-type: none"> 1. Click the Oracle Names Servers folder. 2. Select an Oracle Names Server. 3. Choose Configure Server from the drop-down list box. 4. Click the Advanced tab. 5. Specify the log directory and file name.

3. Choose Save Network Configuration from the File menu.
4. Choose Exit from the File menu.

The Net8 Assistant application exits.

Setting Log Settings During Runtime of Control Utilities

Logging can be set during a runtime of a control utilities. Note that setting logging with a control utility will not set parameters in the *.ORA files; the setting is only valid for the session of the control utility:

- For the listener, use the SET LOG_FILE and SET LOG_DIRECTORY commands from the LSNRCTL control utility, as described in "[Listener Control Utility \(LSNRCTL\)](#)" on page A-3.
- For an Oracle Names server, use the SET LOG_FILE_NAME command from the NAMESCTL control utility, as described in "[Oracle Names Control Utility \(NAMESCTL\)](#)" on page A-31, or set it through the Net8 Assistant:
 - a. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
 - b. Click the Oracle Names Servers folder.
 - c. Select an Oracle Names Server.
 - d. Choose Manage Server from the drop-down list box.
 - e. Click the Logging tab.
 - f. Specify the log directory and file name.

Note: For Oracle Connection Manager, logging can only be set from the Oracle Connection Manager configuration file, as described in "[Oracle Connection Manager Parameters \(CMAN.ORA\)](#)" on page C-76.

- g. Choose Save Network Configuration from the File menu.
 - h. Choose Exit from the File menu.
- The Net8 Assistant application exits.

Using Log Files

To use a log file to diagnose a network error:

1. Review the log file for the most recent error number you received from the application. Note that this is almost always the last entry in the log file.
2. Starting from the bottom of the file, locate the first non-zero entry in the error report. This is usually the actual cause.
3. If that error does not provide the desired information, review the next error in the stack until you locate the correct error information.
4. If the cause of the error is still not clear, turn on tracing and re-execute the statement that produced the error message.

Listener's Log Audit Trail

The listener log file contains Audit Trail information that allows you to gather and analyze network usage statistics, as well as information indicating the following:

- a client connection request
- a start, stop, status, reload or service command issued by the Listener Control Utility

Note that you cannot turn this feature off.

Format of the Listener's Log Audit Trail

The Audit Trail formats text into the following fields: *Timestamp*, *Connect Data*, *Application*, *[Protocol Info]*, *event*, *[SID | service]*, *Return Code*. Properties of the Audit Trail are as follows:

- Each field is delimited by an asterisk (*)
- PROTOCOL INFO and SID | SERVICE appear only when a connection is attempted
- A successful connection or command returns a code of 0
- A failure produces a code that maps to an error message

Typical output to the log file upon a reload request is as follows:

Figure 11-3 Typical Audit Trail Information for Successful Reload Request

```
28-dec-98 14:12:18 * (connect_data=(service=sales.com)(cid=
(program=)(host=sales-pc)(user=system)) (command=reload) (arguments=64)
(service=listener) (version=(version=135282688))* reload * 0
```

Typical output to the log file upon a connection request is as follows:

Figure 11-4 Typical Audit Trail Information for Successful Connection Request

```
28-dec-98 14:16:21 * (connect_data=(service=sales.com)(cid=
(program=c:\ora81\bin\sqlplus.exe)
(host>windowpc)(user=dsteiner))) * (address=(protocol=tcp)
(host=144.25.23.246)(port=3366))
* establish * sales.com * 0
```

Notice that the user ID is recorded as well as the platform, protocol, and software used to make the connection.

Using Audit Trail Information

You can use Audit Trail information to view trends and user activity by first storing it in a table and then collating it into a report format. To import the data into a table, use an import utility such as SQL*Loader.

Understanding Oracle Connection Manager Logs

The Oracle Connection Manager CMGW gateway process creates a log file called `CMAN_PID.LOG`, and the CMADMIN administrative process creates a log file called `CMADM_PID.LOG`. The log files are located in `$ORACLE_HOME/network/log` on UNIX and `ORACLE_HOME\network\log` on Windows NT.

Figure 11–5 CMAN_PID.LOG

```
(TIMESTAMP=30-DEC-98 18:03:10)(EVENT=10)(VERSION=8.1.4.0.0)
(TIMESTAMP=30-DEC-98 18:03:10)(EVENT=36)(rule_list=
(rule=(src=spcstn)(dst=x)(srv=x)(act=accept)))
(TIMESTAMP=30-DEC-98 18:03:10)(EVENT=32)(PARAMETER_LIST=(MAXIMUM_
RELAYS=1024)(RELAY_STATISTICS=no)(AUTHENTICATION_LEVEL=0)(LOG_LEVEL=1)(SHOW_TNS_
INFO=no)(ANSWER_TIMEOUT=0)(MAXIMUM_CONNECT_DATA=1024)(USE_ASYNC_
CALL=yes)(TRACING=no)(TRACE_DIRECTORY=default)(MAX_FREELIST_BUFFERS=0))
(TIMESTAMP=30-DEC-98 18:03:10)(EVENT=34)(ADDRESS_LIST=
(ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1610)(QUEUE_SIZE=32)))
(TIMESTAMP=30-DEC-98 18:03:12)(EVENT=38)(COMMAND=2)
(TIMESTAMP=30-DEC-98
18:03:27)(EVENT=26)(RLYNO=0)(SRC=(ADDRESS=(PROTOCOL=tcp)(HOST=spcstn.us.oracle.c
om)(PORT=34758)))(DST=(ADDRESS=(PROTOCOL=tcp)(HOST=144.25.187.89)(PORT=1581)))
(TIMESTAMP=30-DEC-98 18:03:43)(EVENT=28)(RLYNO=0)(SINCE=30-DEC-98
18:03:27)(STATISTICS=(IN=(BYTES=0)(PACKETS=0)(DCDS=0)(OUT=(BYTES=0)(PACKETS=0)(D
CDS=0)))
```

Figure 11–6 CMADM_PID.LOG

```
(TIMESTAMP=30-DEC-98 18:03:09)(EVENT=Sent Admin Status to UI)
(TIMESTAMP=30-DEC-98 18:03:10)(EVENT=CMan Registration)
```

CMAN_PID.LOG Event Codes

The CMAN_PID.LOG reports events using event codes numbers. This section explains what each of these event codes represents dependent on the log level that was The event code reported is dependent upon the log level set with the LOG_LEVEL parameter in CMAN.ORA:

Log Level 1 Events

Code	Description
10	Gateway is starting up
12	Gateway is shutting down
14	Listening on TNS address(es)
18	Answer failed Additional Information: See "Reasons for Event Code 18" on page 11-29.
20	Refusing in-coming call Additional Information: See "Reasons for Event Code 20" on page 11-29.
26	Relay is now open
28	Relay is now closed
30	Statistics report
32	<parameter_list>
34	<address_list>
36	<parameter_list>
38	CMCTL command
40	CMCTL command refused because the gateway is busy
42	Dead connection detected
44	Relay has timed out
11	Bad <address_list> argument
13	Bad <parameter_list> argument
15	Bad <rule_list> argument
23	Bad CMCTL record
25	Command line argument is too long
27	Memory allocation failure

Code	Description
29	TNS error
31	TNS error doing CMCTL

Reasons for Event Code 18 The answer can fail due to the following:

Code	Description
1	Timed out
2	Connect data buffer is too small
3	Refused by TNS
4	TNS packet checksum error

Reasons for Event Code 20 The incoming call can be refused if:

Code	Description
1	Gateway is shutting down
3	No connect data on in-coming call
4	Bad connect data on in-coming call
5	All relays are in use
6	Unable to get relay buffers
7	Fatal TNS error
8	No available ASO service
9	Reject" from filter rule
10	Out-going call failed
11	Refused by Net8/TNS
12	Listener is not running
13	Listener is not reachable
14	Host name lookup failure
15	Protocol adapter (and probably the protocol stack) not loaded
16	No SOURCE_ROUTE set in TNSNAMES.ORA or Oracle Names server
17	"reject" from rule or bad connect string data

Log Level 2 Events

Code	Description
102	Answering in-coming call
104	Making out-going call
105	Accepting in-coming call
106	Rule match report

Log Level 3 Events

Code	Description
202	Call will block (no asynchronous TNS support)
204	Relay blocked Additional Information: See " Reasons for Event Code 204 ".
206	Buffer contains leftover data

Reasons for Event Code 204

Code	Description
1	Waiting for writer to be ready
2	Waiting for writer to clear backlog
3	EWOULDBLOCK error on recovery
4	TNS packet checksum error
5	EWOULDBLOCK or EPARTIAL on send

Log Level 4 Events

Code	Description
302	Read this many bytes
304	Wrote this many bytes
306	Wrote this many bytes on retry

Tracing Error Information

Tracing produces a detailed sequence of statements that describe network events as they are executed. Tracing an operation allows you to obtain more information on the internal operations of the components of Net8 than is provided in a log file. This information is output to files that can be evaluated to identify the events that led to an error.

CAUTION: Tracing uses a large amount of disk space and may have a significant impact upon system performance. Therefore, you should enable tracing only when necessary

This section covers the following topics:

- [Trace File Names](#)
- [Understanding and Setting Trace Parameters](#)
- [Setting Trace Settings During Runtime of Control Utilities](#)
- [Evaluating Net8 Traces](#)
- [Using the Trace Assistant to Examine Your Trace Files](#)

Trace File Names

Each Net8 component produces its own trace file. The table below provides the default file names and a description of the information they contain:

Log File	Contains Error Information about the...
SQLNET.TRC	Client
SVR_PID.TRC	Server
LISTENER.TRC	Listener
NAMES.TRC	Oracle Names Server
CMAN_PID.TRC	Oracle Connection Manager CMGW gateway process
CMADM_PID.TRC	Oracle Connection Manager CMADMIN administrative process

Understanding and Setting Trace Parameters

Parameters that control tracing, including the type and amount of information logged, as well as the location where the files are stored, are set in the configuration file of each network component as follows:

These trace parameters corresponding to the...	...are set in the following Configuration Files
client	SQLNET.ORA
server	SQLNET.ORA
listener	LISTENER.ORA
Oracle Names Server	NAMES.ORA
Oracle Connection Manager processes	CMAN.ORA

This sections cover the following topics:

- [SQLNET.ORA Parameters](#)
- [LISTENER.ORA Parameters](#)
- [NAMES.ORA Parameters](#)
- [CMAN.ORA Parameters](#)
- [Setting Trace Parameters in Configuration Files](#)

For further information about the parameters described in this section, see [Appendix C, "Configuration Parameters"](#).

SQLNET.ORA Parameters

The following parameters settings can be set in SQLNET.ORA:

SQLNET.ORA Parameter	Net8 Assistant Option	Description
TRACE_LEVEL_CLIENT	Client Information: Trace Level	<p>Indicates the level of detail the trace facility records for the client. The trace level value can either be a value within the range of 0 to 16 (where 0 is no tracing and 16 represents the maximum amount of tracing) or a value of OFF, ADMIN, USER, or SUPPORT.</p> <ul style="list-style-type: none"> ■ OFF (equivalent to 0) provides no tracing. ■ USER (equivalent to 4) traces to identify user-induced error conditions. ■ ADMIN (equivalent to 6) traces to identify installation-specific problems. ■ SUPPORT (equivalent to 16) provides trace information for troubleshooting information for support.
TRACE_DIRECTORY_CLIENT	Client Information: Trace Directory	<p>Establishes the destination directory for the client trace file. By default, the client directory is <code>\$ORACLE_HOME/network/trace</code> on UNIX and <code>ORACLE_HOME\network\trace</code> on Windows NT.</p>

SQLNET.ORA Parameter	Net8 Assistant Option	Description
TRACE_FILE_CLIENT	Client Information: Trace File	Sets the name of the log file for the client. By default the log name is SQLNET.TRC.
TRACE_UNIQUE_CLIENT	Client Information: Unique Trace File Name	Determines whether or not a unique trace file is created. By default, the value is OFF. The same trace file name is used for every client. If the value is OFF, when a new trace file is created for a client, it overwrites the existing file. If the value is set to ON, a process identifier is appended to the name of each trace file generated so that several can coexist. This parameter creates unique trace files named SQLNETPID.TRC
TRACE_LEVEL_SERVER	Server Information: Trace Level	Indicates the level of detail the trace facility records for the server. The trace level value can either be a value within the range of 0 to 16 (where 0 is no tracing and 16 represents the maximum amount of tracing) or a value of OFF, ADMIN, USER, or SUPPORT. <ul style="list-style-type: none"> ■ OFF (equivalent to 0) provides no tracing. ■ USER (equivalent to 4) traces to identify user-induced error conditions. ■ ADMIN (equivalent to 6) traces to identify installation-specific problems. ■ SUPPORT (equivalent to 16) provides trace information for troubleshooting information for support.
TRACE_DIRECTORY_SERVER	Server Information: Trace Directory	Establishes the destination directory for the server trace file. By default, the server directory is \$ORACLE_HOME/network/trace on UNIX and ORACLE_HOME\network\trace on Windows NT.
TRACE_FILE_SERVER	Server Information: Trace File	Sets the name of the trace file for the client. By default the log name is SVR_PID.TRC.

You can also manually add the following optional tracing parameters for the TNSPING utility to SQLNET.ORA. TNSPING determines whether or not a service (such as a database, an Oracle Names Server, or other TNS services) on a Net8 network can be successfully reached.

SQLNET.ORA Parameter	Description
TNSPING.TRACE_LEVEL	<p>Indicates the level of detail the trace facility records for the TNSPING utility. The trace level value can either be a value within the range of 0 to 16 (where 0 is no tracing and 16 represents the maximum amount of tracing) or a value of OFF, ADMIN, USER, or SUPPORT.</p> <ul style="list-style-type: none">■ OFF (equivalent to 0) provides no tracing.■ USER (equivalent to 4) traces to identify user-induced error conditions.■ ADMIN (equivalent to 6) traces to identify installation-specific problems.■ SUPPORT (equivalent to 16) provides trace information for troubleshooting information for support.
TNSPING.TRACE_DIRECTORY	<p>Establishes the destination directory for TNSPING trace file. By default, the directory is <code>\$ORACLE_HOME/network/trace</code> on UNIX and <code>ORACLE_HOME\network\trace</code> on Windows NT.</p>

LISTENER.ORA Parameters

The following trace parameters can be set in LISTENER.ORA:

LISTENER.ORA Parameter	Net8 Assistant Option	Description
<i>TRACE_LEVEL_listener_name</i>	Trace Level	Indicates the level of detail the trace facility records for the listener. The trace level value can either be a value within the range of 0 to 16 (where 0 is no tracing and 16 represents the maximum amount of tracing) or a value of ADMINISTRATOR, USER, or SUPPORT. <ul style="list-style-type: none">▪ USER (equivalent to 4) traces to identify user-induced error conditions.▪ ADMINISTRATOR (equivalent to 6) traces to identify installation-specific problems.▪ SUPPORT (equivalent to 16) provides trace information for troubleshooting information for support.
<i>TRACE_DIRECTORY_listener_name</i> <i>TRACE_FILE_listener_name</i>	Trace File	Establishes the destination directory and file for the trace file. By default the directory is <code>\$ORACLE_HOME/network/trace</code> on UNIX and <code>ORACLE_HOME\network\trace</code> on Windows NT, and the file name is LISTENER.TRC.

NAMES.ORA Parameters

The following trace parameters can be set in NAMES.ORA:

NAMES.ORA Parameter	Net8 Assistant Option	Description
NAMES.TRACE_DIRECTORY	Trace Directory	Establishes the destination directory for trace files. By default, the directory is <code>\$ORACLE_HOME/network/trace</code> on UNIX and <code>ORACLE_HOME\network\trace</code> on Windows NT.
NAMES.TRACE_FILE	Trace File	Sets the name of the trace file. By default the trace name is NAMES.TRC.
NAMES.TRACE_UNIQUE	Make Trace File Unique	If set to ON, creates a unique file name for each trace session, allowing multiple trace files to coexist. A process identifier is appended to the name of the trace file generated. This parameter creates unique trace files named <code>NAMESPID.TRC</code> .

CMAN.ORA Parameters

The following trace parameters can be set in CMAN.ORA:

CMAN.ORA Parameter	Description
TRACING	Sets tracing ON or OFF.
TRACE_DIRECTORY	Establishes the destination directory for trace file. By default, the directory is <code>\$ORACLE_HOME/network/trace</code> on UNIX and <code>ORACLE_HOME\network\trace</code> on Windows NT.

Setting Trace Parameters in Configuration Files

SQLNET.ORA, LISTENER.ORA and NAMES.ORA logging parameters can be set with the Net8 Assistant. CMAN.ORA logging parameter must be set manually, as described in "[Oracle Connection Manager Parameters \(CMAN.ORA\)](#)" on page C-76. To set logging parameters:

1. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
2. Specify the trace parameters:

For this trace file...	Set tracing parameters here...
SQLNET.TRC (for the client)	<ol style="list-style-type: none"> 1. Click the Profile icon. 2. Choose General from the drop-down list box.
SVR_PID.TRC (for the server)	<ol style="list-style-type: none"> 3. Click the Tracing tab. 4. Specify the settings:
LISTENER.TRC	<ol style="list-style-type: none"> 1. Click the Listeners folder. 2. Select a listener. 3. Choose General Parameters from the drop-down list box. 4. Click the Logging and Tracing tab. 5. Specify the settings:
NAMES.TRC	<ol style="list-style-type: none"> 1. Click the Oracle Names Servers folder. 2. Select an Oracle Names server. 3. Choose Configure Server from the drop-down list box. 4. Click the Logging tab. 5. Specify the settings.

3. Choose Save Network Configuration from the File menu.
 4. Choose Exit from the File menu.
- The Net8 Assistant application exits.

Setting Trace Settings During Runtime of Control Utilities

Tracing can be set during a runtime of a control utilities. Note that setting tracing with a control utility will not set parameters in the *.ORA files; the setting is only valid for the session of the control utility:

- For the listener, use the SET TRC_FILE, SET TRC_DIRECTORY and SET TRC_LEVEL commands from the LSNRCTL control utility, as described in "[Listener Control Utility \(LSNRCTL\)](#)" on page A-3.
- For an Oracle Names server, use the SET TRACE_FILE_NAME and SET TRACE_LEVEL command from the NAMESCTL control utility, as described in "[Oracle Names Control Utility \(NAMESCTL\)](#)" on page A-31, or set it through the Net8 Assistant:
 - a. Start the Net8 Assistant:
 - On UNIX, run `netasst` at `$ORACLE_HOME/bin`.
 - On Windows NT, choose Start > Programs > Oracle - *HOME_NAME* > Network Administration > Net8 Assistant.
 - b. Click the Oracle Names Servers folder.
 - c. Select an Oracle Names Server.
 - d. Choose Manage Server from the drop-down list box.
 - e. Click the Logging tab.
 - f. Specify the trace level, directory and file name.

Note: For Oracle Connection Manager, tracing can only be set from the Connection Manager configuration file, as described in "[Oracle Connection Manager Parameters \(CMAN.ORA\)](#)" on page C-76.

- g. Choose Save Network Configuration from the File menu.
- h. Choose Exit from the File menu.

The Net8 Assistant application exits.

Evaluating Net8 Traces

Evaluating trace files either manually, or by using the Trace Assistant tool will help you to diagnose and troubleshoot network problems by giving you a better understanding of the following:

- the flow of packets between network nodes
- which component of Net8 is failing
- pertinent error codes

Understanding the Flow of Data Packets Between Network Nodes

Net8 performs its functions by sending and receiving data packets. By specifying a trace level of SUPPORT, you can view the actual contents of the Net8 packet in your trace file. The order of the packet types sent and received will help you to determine how your connection was established.

Understanding Data Packet Formats Each line in the trace file begins with a procedure followed by a message. Following each procedure is a line of hexadecimal data representing actual data. The actual data that flows inside the packet is sometimes viewable to the right of the hexadecimal data.

Below is a list of the Net8 packet keywords and describes the types of packets they represent:

Keyword	Packet Type
NSPTCN	Connect
NSPTAC	Accept
NSPTRF	Refuse
NSPTRS	Resend
NSPDA	Data
NSPCNL	Control
NSPTMK	Marker

Note: This data is not viewable if you are using encryption through an Oracle network product or through EBCDIC data.

For example, the following line describes a procedure called "nscn" sending a NSPTCN packet over the network:

```
nscn: sending NSPTCN packet
```

Each packet has a keyword that denotes the packet type. All packet types begin with the prefix "NSP". It is helpful to remember this when reviewing trace files for specific packet information

Figure 11-7 provides typical packet information:

Figure 11-7 Packet Information

```
nscn: entry
nscn: doing connect handshake...
nscn: sending NSPTCN packet
npsend: entry
npsend: plen=187, type=1
npsend: 187 bytes to transport
npsend:packet dump
npsend:00 BB 00 00 01 00 00 00 |.....|
npsend:01 33 01 2C 0C 01 08 00 |.3.,....|
npsend:7F FF 7F 08 00 00 00 01 |.....|
npsend:00 99 00 22 00 00 08 00 |..."....|
npsend:01 01 28 44 45 53 43 52 |..(DESCR|
npsend:49 50 54 49 4F 4E 3D 28 |PTION=(|
npsend:43 4F 4E 4E 45 43 54 5F |CONNECT_|
npsend:44 41 54 41 3D 28 53 49 |DATA=(SI|
npsend:44 3D 61 70 33 34 37 64 |D=ap347d|
npsend:62 31 29 28 43 49 44 3D |b1)(CID=|
npsend:28 50 52 4F 47 52 41 4D |(PROGRAM|
npsend:3D 29 28 48 4F 53 54 3D |=)(HOST=|
npsend:61 70 32 30 37 73 75 6E |ap207sun|
npsend:29 28 55 53 45 52 3D 6D |)(USER=m|
npsend:77 61 72 72 65 6E 29 29 |warren))|
npsend:29 28 41 44 44 52 45 53 |)(ADDRES|
npsend:53 5F 4C 49 53 54 3D 28 |S_LIST=(|
npsend:41 44 44 52 45 53 53 3D |ADDRESS=|
npsend:28 50 52 4F 54 4F 43 4F |(PROTOCO|
npsend:4C 3D 74 63 70 29 28 48 |L=tcp)(H|
npsend:4F 53 54 3D 61 70 33 34 |OST=ap34|
npsend:37 73 75 6E 29 28 50 4F |7sun)(PO|
npsend:52 54 3D 31 35 32 31 29 |RT=1521)|
npsend:29 29 29 00 00 00 00 00 |)).....|
npsend: normal exit
nscn: exit (0)
```

Understanding Pertinent Error Output

Every time a problem occurs with the connection in Net8, the error code is logged in the trace file with the prefix of <ERROR> or <FATAL>. [Figure 11-8](#) depicts typical trace file error output.

Figure 11-8 Trace File Error Output

```
nspsend: entry
nspsend: plen=244, type=6
ntpwr: entry
ntpwr: exit
-<ERROR>- nspsend: transport write error
nspsend: error exit
nserror: entry
-<ERROR>- nserror: nsres: id=0, op=65, ns=12541, ns2=12560; nt[0]=511,
nt[1]=61,nt[2]=0
-<ERROR>- nsopen: unable to open transport
nricdt: Call failed...
nricdt: exit
-<ERROR>- osnqper: error from nricall
-<ERROR>- osnqper: nr err code: 12203
-<ERROR>- osnqper: ns main err code: 12541
-<ERROR>- osnqper: ns (2) err code: 12560
-<ERROR>- osnqper: nt main err code: 511
-<ERROR>- osnqper: nt (2) err code: 61
-<ERROR>- osnqper: nt OS err code: 0
osnqme: entry
osnqme: reporting nr (1) error: (12203) as rdbms err (12203)
osnqme: exit
-<ERROR>- onstns: Couldn't connect, returning 12203
nricall: Exiting NRICALL with following termination result -1
nricall: exit
osnqme: entry
osnqme: reporting nr (1) error: (12203) as rdbms err (12203)
osnqme: exit
-<ERROR>- onstns: Couldn't connect, returning 12203
-<ERROR>- osnqper: error from nricall
```

The most efficient way to evaluate error codes is to find the most recent NS error code logged. This is because the session layer controls the connection. The most important error messages are the ones at the bottom of the file. They are the most recent errors and the source of the problem with your connection.

For information about the specific return codes, use the Oracle UNIX error tool `oerr`. Use the `oerr` tool to discover more information about Net8 return codes, by entering the following at any command line prompt:

```
oerr tns error_number
```

Using the Trace Assistant to Examine Your Trace Files

Net8 provides a tool called the Trace Assistant to help you understand the information provided in your trace files by converting existing lines of trace file text into a more readable paragraph. Note that the Trace Assistant runs against only a level 16 (SUPPORT) SQL*Net or Net8 trace file.

To run the Trace Assistant, enter the following at any command line prompt:

```
trcasst [options] filename
```

The options are described below.

Option	Description
-o	Displays connectivity and Two Task Common (TTC) information. After the -o the following options may be used: <ul style="list-style-type: none"> ■ c (for summary connectivity information) ■ d (for detailed connectivity information) ■ u (for summary TTC information) ■ t (for detailed TTC information) ■ q (displays SQL commands enhancing summary TTC information)
-p	Oracle Internal Use Only
-s	Displays statistical information
-e	Enables display of error information After the -e, zero or one error decoding level may follow: <ul style="list-style-type: none"> ■ 0 or nothing (translates the NS error numbers dumped from the <code>nerror</code> function plus lists all other errors) ■ 1 (displays only the NS error translation from the <code>nerror</code> function) ■ 2 (displays error numbers without translation)

If no options are provided, then the default is `-odt -e -s`, providing detailed connectivity, detailed Two-Task Common, error level 0, and statistics.

Figure 11-9 depicts how Trace Assistant converts trace file information into a more readable format.

Figure 11-9 Typical Trace Assistant Conversion

Trace File	Converted by Trace Assistant with option -e0 or -e1
nsc2addr: normal exit nsopen: entry nsmal: 404 bytes at 0x10d5a48 nsopen: opening transport... -<ERROR>- ntus2err: sd=13, op=1, resnt[0]=511, resnt[1]=2, resnt[2]=0 -<ERROR>- nserror: nsres: id=0, op=65, ns=12541, ns2=12560; nt[0]=511, nt[1]=2, nt[2]=0 -<ERROR>- nsopen: unable to open transport	Error found. Error Stack follows: id: 00000 Operation code: 00065 NS Error 1: 12541 NS Error 2: 12560 NT Generic Error: 00511 Protocol Error: 00146 OS Error: 00000 NS & NT Errors Translation 12541, 00000, "TNS:no listener" // *Cause: The connection request could not be completed because the listener // is not running. // *Action: Ensure that the supplied destination address matches one of // the addresses used by the listener - compare the TNSNAMES.ORA entry // with // the appropriate LISTENER.ORA file (or TNSNAV.ORA if the connection // is to // go by way of an Interchange). Start the listener on the remote machine. / 12560, 00000, "TNS:protocol adapter error" // *Cause: A generic protocol adapter error occurred. // *Action: Check addresses used for proper protocol specification. Before // reporting this error, look at the error stack and check for lower level // transport errors. For further details, turn on tracing and re-execute the // operation. Turn off tracing when the operation is complete. / 00511, 00000, "No listener" // *Cause: The connect request could not be completed because no // application // is listening on the address specified, or the application is unable to // service the connect request in a sufficiently timely manner. // *Action: Ensure that the supplied destination address matches one of // the addresses used by the listener - compare the TNSNAMES.ORA entry // with // appropriate LISTENER.ORA file. // Start the listener on the remote machine.

However, other errors may also exist within the trace file that were not logged from the nserror function.

Understanding Information Traversing the Network in Net8 Packets

Trace Assistant also allows you to view data packets from both the Net8 and Two Task Common communication layers. Trace Assistant offers you two options to view these packets:

- summary connectivity (using option -oc)
- detailed connectivity (using option -od)

Net8 Packet Examples

The following examples depict how Trace Assistant presents various packets as they are sent to and from the Net8 layer in a variety of transactions:

- bequeathed connection
- redirected connection
- data packet

Note that the packets being sent or received have a prefix of "---> Send *nnn* bytes" or "<--- Received *nnn* bytes" showing that this node is sending or receiving a packet of a certain type and with *nnn* number of bytes. This prefix enables you to determine if the node is the client or the server. The connection request is always sent by the client, but received by the server (or listener).

Figure 11–10 Summary Data Packets Sent in a Bequeathed Connection

Using trcasst -oc <filename>

This example shows two packets. The first is the connect packet that is sent from the client to the listener. The second is the accept packet coming back from the server.

```

---> Send      192 bytes - Connect packet
      Connect data length: 142
(description=(address=(protocol=tcp)(host=dlsun)(port=1521))(connect_
data=(sid=db1)(cid=(program=)(host=dlsun)(user=use1))))
<--- received 24 bytes - accept packet
      Accept data length: 0

```

Figure 11–11 Detailed Data Packets Sent in a Bequeathed Connection

Using `trcasst -od <filename>`

This example shows all of the details sent along with the connect data in negotiating a connection.

```
---> Send      50 bytes - Connect packet
Current NS version number is: 309.
Lowest NS version number can accommodate is: 300.
Global options for the connection:
    can receive attention
    no attention processing
    Don't care
    Maximum SDU size: 2048
    Maximum TDU size: 5120
    NT protocol characteristics:
        Test for more data
        Spawner is running
        Hang on to Listener connection
        Full duplex I/O
        Urgent data support
        Generate SIGURG signal
        Handoff connection to another
    Line turnaround value: 0
    Connect data length: 234
    Connect data offset: 50
    Connect data maximum size: 2048
        Native Services wanted
        Native Services wanted
Cross facility item 1: 0
    Cross facility item 2: 0
    Connection id: 0x0000000000000000
    Packet data is in the following data packet
---> Send      244 bytes - Data packet
(DESCRIPTION=(ADDRESS=(PROTOCOL=beq)(PROGRAM=/private/oracle/bin/oracle)(ARGV0=oracle)(A
RGS='(DESCRIPTION=(LOCAL=YES) (ADDRESS=(PROTOCOL=beq)))(DETACH=NO))(CONNECT_
DATA=(CID=(PROGRAM=(HOST=dlsun)(USER=usel))))))
<--- Received 24 bytes - Accept packet
    Accepted NS version number is: 307.
Global options for the connection:
    no attention processing
    Don't care
    Accepted maximum SDU size: 2048
    Accepted maximum TDU size: 4096
    Connect data length: 0
        Native Services wanted
        Native Services wanted
```

Figure 11–12 Summary Data Packets Sent in a Redirected Connection**Using `trcasst -oc <filename>`**

```

---> Send      187 bytes - Connect packet
      Connect data length: 153
      (DESCRIPTION=(CONNECT_
DATA=(SID=ap347dbl)(CID=(PROGRAM=(HOST=apsun)(USER=use2)))(ADDRESS_
LIST=(ADDRESS=(PROTOCOL=tcp)(HOST=apsun)(PORT=1521))))
<--- Received 8 bytes - Resend packet
---> Send      187 bytes - Connect packet
      Connect data length: 153
      (DESCRIPTION=(CONNECT_DATA=(SID=apdbl)(CID=(PROGRAM=(HOST=apsun)(USER=use2)))(ADDRESS_
LIST=(ADDRESS=(PROTOCOL=tcp)(HOST=apsun)(PORT=1521))))
<--- Received 24 bytes - Accept packet
      Accept data length: 0

```

Figure 11–13 Data Packet**Using `trcasst -oc <filename>` or `-od <filename>`**

Once the connection is established, data is given to Net8 from the Two-Task Common layer to be sent to the other node. Both summary and detailed views yield the same summary information.

```

---> Send      30 bytes - Data packet
<--- Received 201 bytes - Data packet
---> Send      439 bytes - Data packet
<--- Received 400 bytes - Data packet

```

Two Task Common Packet Examples

Two-Task Common handles requests such as open cursor, select rows, and update rows that are directed to the database. All requests are answered by the server. If you request to logon, a response is returned from the database that the request was completed. [Figure 11-14](#) on page 11-48 through [Figure 11-16](#) on page 11-51 show the type of information you can expect.

Summary information for Two-Task Common is different from other displays in that it shows two packets on each line, rather than one. This is done to mirror the request/response pairings process by which Two-Task Common operates.

Figure 11-14 Two Task Common Summary Information

Using `trcasst -ou <filename>`

(O3LOGA)	1st half of challenge-response logon	80	78
(O3LOGON)	2nd half of challenge-response logon	97	59
(OOPEN)	# 1	21	16
(OPARSEX)	# 1	245	59
(OCLOSE)	# 1	17	11
(OVERSION)		29	16
(OOPEN)	# 2	21	16

On each line, the first item displayed is the actual request made. The second item is a cursor number, if one is involved with the transaction. The third item is either a listing of the flags or the SQL command that is being answered. The flag indicates that a request has the following characteristics:

!PL/SQL = Not a PL/SQL request

COM = Commit

IOV = Get I/O Vector

DEFN = Define

EXEC = Execute

FETCH = Fetch

CAN = Cancel

DESCSEL = Describe select

DESCBND = Describe Bind

BND = Bind

PARSE = Parse

EXACT = Exact

The number of bytes sent and received are displayed at the far right.

The OOPEN on line three is a prime example of how the output displays the request/response pairs. The OOPEN appears with a #1 following it indicating that an "Open cursor" request was sent from the client and the server responded with the cursor number 1 that it opened. Because a request/response pairing is placed on one line, you should not combine this option with any of the connectivity options.

Figure 11-14 Two Task Common Summary Information**Using trcasst -ou <filename>**

(OALL7)	# 2	Parse Can Defn=2 Exec Fetch "SELECT A.V	268	100
(OOPEN)	# 3		21	16
(OALL7)	# 3	Parse Exec=1 "SELECT USER FROM SYS.DUAL	152	70
(OALL7)	# 3	Defn=1 Fetc	117	88
(OCLOSE)	# 3		17	11

On each line, the first item displayed is the actual request made. The second item is a cursor number, if one is involved with the transaction. The third item is either a listing of the flags or the SQL command that is being answered. The flag indicates that a request has the following characteristics:

!PL/SQL = Not a PL/SQL request

COM = Commit

IOV = Get I/O Vector

DEFN = Define

EXEC = Execute

FETCH = Fetch

CAN = Cancel

DESCSEL = Describe select

DESCBND = Describe Bind

BND = Bind

PARSE = Parse

EXACT = Exact

The number of bytes sent and received are displayed at the far right.

The OOPEN on line three is a prime example of how the output displays the request/response pairs. The OOPEN appears with a #1 following it indicating that an "Open cursor" request was sent from the client and the server responded with the cursor number 1 that it opened. Because a request/response pairing is placed on one line, you should not combine this option with any of the connectivity options.

Figure 11–15 Two Task Common Summary Information**Using trcasst -ot <filename>**

```
start of user function (TTIFUN)
    1st half of challenge-response logon (O3LOGA)
        Username: applsys
        Terminal: ttyp5
        Machine: ap207sun
        System User: mwarren
        Process: 24459
        Program: aiap45@ap207sun (TNS interface)
return opi parameter (TTIRPA)
    OPI parameter: 3309B1A977A62A3C
start of user function (TTIFUN)
    2nd half of challenge-response logon (O3LOGON)
        Username: applsys
        Terminal: ttyp5
        Machine: ap207sun
        System User: mwarren
        Process: 24459
        Program: aiap45@ap207sun (TNS interface)
ORACLE function complete (TTIOER)
start of user function (TTIFUN)
    Open a cursor
return opi parameter (TTIRPA)
    Cursor #: 1
start of user function (TTIFUN)
    Parse and Execute (OPARSEX) Cursor # 1
alter session set nls_language= 'AMERICAN' nls_territory= 'AMERICA' nls_currency=
'$' nls_iso_currency= 'AMERICA' nls_numeric_characters= '.,' nls_date_format=
'DD-MON-YY' nls_date_language= 'AMERICAN' nls_sort= 'BINARY'
ORACLE function complete (TTIOER)
start of user function (TTIFUN)
    Close cursor (OCLOSE) Cursor # 1
V6 Oracle func complete (TTISTA)
    Succeeded
```

Figure 11-16 Detailed SQL information on top of summary Two-Task**Using trcasst -ouq <filename>**

(O3LOGA)	1st half of challenge-response logon	180	78
(O3LOGON)	2nd half of challenge-response logon	197	59
(OOPEN)	# 1	21	16
(OPARSEX)	# 1 alter session set nls_language= 'AMERICAN' nls_ territory= 'AMERICA' nls_currency= '\$' nls_iso_ currency= 'AMERICA' nls_numeric_characters= '.,' nls_ date_format= 'DD-MON-YY' nls_date_language= 'AMERICAN' nls_sort= 'BINARY'	245	59
(OCLOSE)	# 1	17	11
(O71SESOPN)	(get session ID)	47	18
(OOPEN)	# 1	21	16
(OVERSION)	Oracle7 Server Release 8.1.3.0.0 - Production Release with the distributed and parallel query optionsPL/SQL Release 2.1.4.0.0 - Production	29	157
(O71SESOPN)	(get session ID)	47	18

Add q to your summary Two-Task Command to display the detailed SQL information given automatically in the detailed Two-Task option.

Analyze the Data Collected into Appropriate Statistics

The type of statistics gathered is on the order of how many calls (TTC), packets and bytes were sent and received between the network partners. The following example depicts typical trace file statistics:

Figure 11–17 Typical Trace File Statistics

Using `trcasst -s <filename>`

```
=====
Trace File Statistics:
-----
SQL*Net:
  Total Calls:      466 sent,      491 received,      423 upi
  Total Bytes:    119214 sent,    86614 received
  Average Bytes:      255 sent,      176 received
  Maximum Bytes:    2048 sent,    2048 received
GRAND TOTAL PACKETS sent:  466 received:  491
```

Trace File Example

The following example shows a full trace file decoded. This example was created using the Oracle client tool SVRMGRL with the request:

```
connect scott/tiger@june
```

The message ORA-12154: TNS:could not resolve service name was displayed on the screen.

Figure 11–18 Trace File Example

Description	Trace File Information
Note Trace level and location of the trace file in the Trace Configuration Information section.	-- TRACE CONFIGURATION INFORMATION FOLLOWS --- New trace stream is "C:\ORANT\network\trace\sqlnet7.trc" New trace level is 16 --- TRACE CONFIGURATION INFORMATION ENDS ---
The Oracle Names component cannot find net service name JUNE.COM. Note client adds ".COM" extension to net service name "JUNE".	nnfotran: tnsname.ora entry for name "june.com" not found nnftqnm: Error querying june.com of attribute A.SMD errcode 406 nnfgrwsp: Query unsuccessful, skipping to next adapter
Client attempts to access a Names Server (oramesrvr0) to resolve net service name address.	nnfgrwsp: Switching to ONAMES adapter nnfgrwsp: Original name: june nnfgrwsp: Qualified name: june.com nngsget_get_stream: looking for "(DESCRIPTION=(CONNECT_ DATA=(RPC=ON))(ADDRESS=(PROTOCOL=tcp)(HOST=oramesrvr0)(PORT=1575)))" nngsget_get_stream: cache miss, opening new stream nngsnad_new_stream_addr: "(DESCRIPTION=(CONNECT_ DATA=(RPC=ON))(ADDRESS=(PROTOCOL= tcp)(HOST=oramesrvr0)(PORT=1575)))" nngsget_get_stream: no caller address will be sent to callee

Figure 11–18 Trace File Example

Description	Trace File Information
Network Routing (NR) performs routing to the Oracle Names server (oranamesrvr0).	<pre> nricall: entry nric2a: entry nric2a: Getting local community information nriglp: entry nriglp: Looking for local addresses setup by nrigla nriglp: No addresses in the preferred address list nriglp: exit nric2a: TNSNAV.ORA is not present. No local communities entry. nrigla: entry nrigla: Getting local address information nrigla: Simple address... nrigla: No community component so just use straight address nrigla: exit nridst: entry nridst: Resolving address to use to call destination or next hop nridst: Found destination address nridst: Local address nridst: Local destination community found nridst: exit nric2a: This is a local community access nric2a: exit nricall: Got routable address information nricall: Making call with following address information: (DESCRIPTION=(CONNECT_ DATA=(RPC=ON))(ADDRESS=(PROTOCOL=tcp)(HOST=oranamesrvr0)(PORT =1575))) nricdt: entry nricdt: Calling with outgoing connect data (DESCRIPTION=(CONNECT_ DATA=(RPC=ON))(ADDRESS=(PROTOCOL=tcp)(HOST=oranamesrvr0)(PORT =1575))) </pre>
Network Session (NS) sets up the session to the Oracle Names server.	<pre> nscall: entry nscall: connecting... nsc2addr: entry nsc2addr: (DESCRIPTION=(CONNECT_ DATA=(RPC=ON))(ADDRESS=(PROTOCOL=tcp)(HOST=oranamesrvr0)(PORT =1575))) </pre>

Figure 11–18 Trace File Example

Description	Trace File Information
Network Transport (NT) sets up the transport session.	<pre> nttbnd2addr: entry nttbnd2addr: port resolved to 1575 nttbnd2addr: looking up IP addr for host: oranamesrvr0 nttbnd2addr: exitnsopen: entry nsmal: entry nsmal: 330 bytes at 0x30d76e74 nsmal: normal exit nsopen: opening transport... nttcon: entry nttcon: toc = 1 nttcnp: entry nttcnp: creating a socket. nttcnp: exit nttcni: entry nttcni: trying to connect to socket 1. ntt2err: entry </pre>
Network Transport (NT) returns the error "no listener" as the Names Server is not running.	<pre> -<ERROR>- ntt2err: soc 1 error - operation=1, ntresnt[0]=511, ntresnt[1]=61 ntresnt[2]=0 ntt2err: exit nttcni: exit nttcon: exit nserror: entry </pre>
The error is propagated to the next layer (NS).	<pre> -<ERROR>- nserror: nsres: id=0, op=65, ns=12541, ns2=12560; nt[0]=511, nt[1]=61,nt[2]=0 -<ERROR>- nsopen: unable to open transport nsmfr: entry nsmfr: 330 bytes at 0x30d76e74 nsmfr: normal exit nsopen: error exit nscall: error exit nricdt: Call failed... nricfg: entry nricfg: exit nricdt: Call made to destination nricdt: exit nricall: Failed to copy originating community name value binding nricall: Exiting NRICALL with following termination result -1 nricall: exit nngsfad_free_stream_addr: "(DESCRIPTION=(CONNECT_ DATA=(RPC=ON))(ADDRESS=(PROTOCOL=tcp)(HOST=oranamesrvr0)(PORT =1575)))" -<ERROR>- nngsget_get_stream: open failure, error stack follows </pre>

Figure 11–18 Trace File Example

Description	Trace File Information
The errors are propagated to the next layer (TNS)	<pre>TNS-12224: TNS:no listener TNS-12541: TNS:no listener TNS-12560: TNS:protocol adapter error TNS-00511: No listener nnfgrwsp: Query unsuccessful, skipping to next adapter</pre>
The address is not found on any Oracle Names Server as no Oracle Names Server is available.	<pre>nnfun2a: address for name "june" not found nngsfad_free_stream_addr: "(DESCRIPTION = (CONNECT_DATA= (RPC=ON)) (ADDRESS=(PROTOCOL=tcp) (HOST=oranamesrvr0) (PORT=1575)))" nngtdei_deinit_msg: free message pool block nngtfms_free_msg: message ID -10429 nngtfms_free_msg: message was a request nngtfms_free_msg: message free, type 100 nngtfoa_free_objarr: free message object array nngtfmt_free_msg_type: type-specific message free, type 100 nngtfoa_free_objarr: free message object array nngtfms_free_msg: message ID 0 nngtfms_free_msg: message was a response nngtfms_free_msg: message free, type 0 nngsdei_deinit_streams: deinit nngscls_close_stream: UID 11 not established, ignored nngscls_close_stream: UID 0 not established, ignored osnqrn: Return code from nnfsn2a is 409</pre>
Error is returned to the user.	<pre>--<ERROR>- onstns: Couldn't connect, returning 12154 onstns: exit osnqtg: Count in the OSN global area is now 0 rignbd: entry nrignbd: exit osnqtg: Count in the NL global area is now 0</pre>

Trace File Example Summary

This trace file provides a summary of what occurs with Net8 when you encounter the error ORA-12154: Could not resolve service name. In this example, a client is unsuccessful in making a connection to net service name JUNE. This is because a NAMES.DEFAULT_DOMAIN = COM parameter exists in the SQLNET.ORA file. This parameter adds the ".COM" extension to all net service names requested, including the net service name JUNE. Unfortunately, this net service name is defined in neither the client's TNSNAMES.ORA file, nor an Oracle Names Server. To troubleshoot this problem, the user should:

- edit the SQLNET.ORA file to remove the NAMES.DEFAULT_DOMAIN configuration parameter; or
- request a connection to JUNE.COM instead of JUNE

Contacting Oracle Customer Support

If you are still unable to resolve your problems or if you are requested to contact Oracle Customer Support to report the error, please have the following information at hand:

- The hardware and operating system release number on which your application(s) is running
- The up-to-five-digit release number of all the Oracle networking products involved in the current problem
- The third-party vendor and version you are using
- If you encountered one or more error codes or messages, the exact code numbers and message texts in the order they appeared.
- The kind of links that exist between the client and server
- A description of what does work
- The exact error message, if there is one
- A Net8 trace, if possible; if not, the log file is sufficient

Part III

Reference

Part III provides reference material for this guide. It contains the following appendices:

- [Appendix A, "Control Utilities for the Listener, Oracle Names Server, and Oracle Connection Manager"](#)
- [Appendix B, "Sample Configuration Files"](#)
- [Appendix C, "Configuration Parameters"](#)

A

Control Utilities for the Listener, Oracle Names Server, and Oracle Connection Manager

Net8 provides you with utilities through which you can control the listener, Oracle Names and Oracle Connection Manager through control utilities. This appendix lists the commands that are available with each utility, including any applicable prerequisites, passwords, syntax or argument rules, usage notes or examples to help you use them.

The three control utilities described in this appendix are:

- [Listener Control Utility \(LSNRCTL\)](#)
- [Oracle Names Control Utility \(NAMECTL\)](#)
- [Oracle Connection Manager Control Utility \(CMCTL\)](#)

Operating Modes

The control utilities operate for the duration of runtime session. You can run utilities in one of three modes:

- Interpreter mode - The control utility is loaded from the control utility command line. When loaded, the program displays a prompt, such as:

```
lsnrctl>
```

- Command line mode - You can also execute most commands from the operating system command line by running the control utility with a complete command as a parameter to the program. In this case, control utility will load and execute the command, then return the operating system prompt. Sample commands are:

```
namesctl start  
namesctl status sales.com
```

- Batch command mode - You can combine commands in a standard text file, then run them as a sequence of commands. To execute in batch mode, use the format:

```
control_utility @file_name
```

- You can use either REM or # to identify comments in the batch script; all other lines are considered commands. Any commands that would typically require confirmation do not require confirmation during batch execution.

Listener Control Utility (LSNRCTL)

The Listener Control Utility (LSNRCTL) manages the listener. The general syntax of the Listener Control Utility is as follows:

```
lsnrctl command [listener_name]
```

where *listener_name* is the name of the listener defined in the LISTENER.ORA file. It is not necessary to identify the listener if you are using the default listener, named LISTENER.

LSNRCTL contains several types of commands:

- Operational commands such as START, STOP, and so forth.
- Modifier commands, such as SET *property*
- Informational commands, such as STATUS, SHOW *property*
- Command utility operational commands such as EXIT, QUIT, and HELP

You can use the any of these utilities to perform basic management functions on one or more listeners. Additionally, you can view and change Oracle Names server parameter settings.

SET and SHOW Modifiers

You can use the modifier SET to change some parameter values for a listener or the LSNRCTL environment during the LSNRCTL control utility session.

Security

If the `PASSWORDS.listener_name` parameter is set in the LISTENER.ORA, the LSNRCTL control utility will require a SET PASSWORD command for any sensitive operations, such as starting or stopping the listener.

LSNRCTL Commands

The following commands are available through the Listener Control Utility (LSNRCTL):

CHANGE_PASSWORD

Purpose:	Allows you to dynamically change the encrypted password of a listener. This will not change unencrypted passwords already established in a listener configuration file. It only establishes a new password, or changes a password that has been encrypted in the listener configuration file.
Prerequisites:	None
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>lsnrctl change_password [listener_name]</code> From the LSNRCTL program: <code>lsnrctl> change_password [listener_name]</code>
Arguments:	[<i>listener_name</i>]
Usage Notes:	The control utility prompts you for your old password, then for the new one. It asks you to re-enter the new one, then changes it. Neither the old nor the new password displays during this procedure.
Example:	<pre>lsnrctl> change_password Old password: New password: Reenter new password: Connecting to (ADDRESS=(PROTOCOL=ipc)(KEY=iris)) Password changed for LISTENER The command completed successfully</pre>

DBSNMP_START

Purpose:	Starts the SNMP subagent for an Oracle database running on the same node.
Prerequisites:	Must be run locally
Password required if one has been set:	No
Syntax:	From the operating system: lsnrctl dbsnmp_start From the LSNRCTL program: lsnrctl> dbsnmp_start
Arguments:	None
Usage Notes:	None
Example:	lsnrctl> dbsnmp_start

DBSNMP_STATUS

Purpose:	Verifies whether the SNMP subagent for an Oracle database is running.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	From the operating system: lsnrctl dbsnmp_status From the LSNRCTL program: lsnrctl> dbsnmp_status
Arguments:	None
Usage Notes:	DBSNMP STATUS must be run on the same node the Oracle database is on.
Example:	lsnrctl> dbsnmp_status

DBSNMP_STOP

Purpose:	Stops the SNMP subagent for an Oracle database running on the same node.
Prerequisites:	Must be run locally
Password required if one has been set:	No
Syntax:	From the operating system: <code>lsnrctl dbsnmp_stop</code> From the LSNRCTL program: <code>lsnrctl> dbsnmp_stop</code>
Arguments:	None
Usage Notes:	None
Example:	<code>lsnrctl> dbsnmp_stop</code>

EXIT

Purpose:	Quits LSNRCTL, and returns you to the operating system prompt.
Prerequisites:	None
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>lsnrctl exit</code> From the LSNRCTL program: <code>lsnrctl> exit</code>
Arguments:	None
Usage Notes:	This command is identical to QUIT
Example:	<code>lsnrctl> exit</code>

HELP

Purpose:	Provides a list of all the LSNRCTL commands available.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	From the operating system: <code>lsnrctl help [command]</code> From the LSNRCTL program: <code>lsnrctl> help [command]</code>
Arguments:	help command
Example:	<code>lsnrctl> help</code> The following operations are available An asterisk (*) denotes a modifier or extended command: <code>change_password</code> <code>dbstmp_start</code> <code>dbstmp_stop</code> <code>dbstmp_status</code> <code>exit</code> <code>quit</code> <code>reload</code> <code>services</code> <code>set*</code> <code>show*</code> <code>spawn</code> <code>start</code> <code>status</code> <code>stop</code> <code>trace</code> <code>version</code>

QUIT

Purpose:	Quits LSNRCTL, and returns you to the operating system prompt.
Prerequisites:	None
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>lsnrctl quit</code> From the LSNRCTL program: <code>lsnrctl> quit</code>
Arguments:	None
Usage Notes:	This command is identical to EXIT
Example:	<code>lsnrctl> quit</code>

RELOAD

Purpose:	Shuts down everything except listener addresses, and re-reads the LISTENER.ORA file. This command enables you to add or change services without actually stopping the listener.
Prerequisites:	This will not work on valid nodes. In this case, the listener must be stopped and restarted.
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>lsnrctl reload [listener_name]</code> From the LSNRCTL program: <code>lsnrctl> reload [listener_name]</code>
Arguments:	listener name

RELOAD

Usage Notes: If there are any passwords in the listener configuration file, you must use the SET PASSWORD command before you can use the RELOAD command.

You must set the password from within the LSNRCTL program; you cannot set it from the operating system command line. The method for setting the password depends on whether you are using the encrypted password feature. If you are *not* using an encrypted password, enter the password on the LSNRCTL command line.

Example: `lsnrctl> reload`

SAVE_CONFIG

Purpose: Compares the current configuration state of the listener, including trace level, trace file, trace directory, logging, and connect timeout, to the LISTENER.ORA file. Any changes are stored in the LISTENER.ORA, preserving formatting, comments, and case as much as possible. Prior to modification of the LISTENER.ORA, a backup of LISTENER.ORA, called LISTENER.BAK, is created.

Password required if one has been set: Yes

If a password is set, the SET PASSWORD command must be issued prior to this command.

Syntax: From the operating system:

```
lsnrctl save_config [listener_name]
```

From the LSNRCTL program:

```
lsnrctl> save_config [listener_name]
```

Arguments: listener name

Usage Notes: This is used by an administrator to save all on-line configuration changes to the listener configuration file. This is especially useful for saving changed encrypted passwords.

Example: `lsnrctl> save_config listener`

SERVICES

Purpose:	Provides detailed information about the services the listener listens for. For example, how many connections have been established, how many refused. It displays three different types of services (dedicated servers from LISTENER.ORA, dispatcher information, and prespawned shadows.)
Prerequisites:	None
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>lsnrctl services [listener_name]</code> From the LSNRCTL program: <code>lsnrctl> services [listener_name]</code>
Arguments:	listener name
Usage Notes:	This is used by a database administrator to get information about the services of the listener. The SET DISPLAY and SET RAWMODE commands change the format of the output.

SERVICES

Example:

```
lsnrctl> services listener_name
```

The output of a LSNRCTL SERVICES command follows:

```
Connecting to (ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC))
```

```
Services Summary...
```

```
PLSExtProc          has 1 service handler(s)
```

```
  DEDICATED SERVER  established:0 refused:0
```

```
  LOCAL SERVER
```

```
db1                  has 3 service handler(s)
```

```
  DEDICATED SERVER  established:0 refused:0
```

```
  LOCAL SERVER
```

```
  DISPATCHER        established:0 refused:0 current:0 max:254
```

```
state:ready
```

```
  D000 <machine: dlsun1770, pid: 1289>
```

```
(ADDRESS=(PROTOCOL=tcp)(DEV=14)(HOST=144.25.17.193)(PORT=2222))
```

```
  DISPATCHER        established:0 refused:0 current:0 max:254
```

```
state:ready
```

```
  D001 <machine: dlsun1770, pid: 1291>
```

```
(ADDRESS=(PROTOCOL=tcp)(DEV=14)(HOST=144.25.17.193)(PORT=2223))
```

```
The command completed successfully
```

SET

Purpose:	Lists the parameter values that can be set using the SET command.
Prerequisites:	None
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>lsnrctl set [command]</code> From the LSNRCTL program: <code>lsnrctl> set [command]</code>
Arguments:	<code>set command</code>
Usage Notes:	You must have set a valid password to be able to use this command if one is listed in the listener configuration parameter, <code>PASSWORDS_listener_name</code> . If there are any passwords in the listener configuration file, you must use the SET PASSWORD command before you can use the SET command. You must set the password from within the LSNRCTL program; you cannot set it from the operating system command line. The method for setting the password depends on whether you are using the encrypted password feature. If you are <i>not</i> using an encrypted password, enter the password on the LSNRCTL command line.

SET

Example:

```
lsnrctl> set
```

The following operations are available after set.

```
connect_timeout  
current_listener  
displaymode  
log_file  
log_directory  
log_status  
password  
raw_mode  
save_config_on_stop  
startup_waittime  
trc_file  
trc_directory  
trc_level  
use_plugandplay
```

SET CONNECT_TIMEOUT

Purpose:	Determines the amount of time the listener will wait for a valid connection request after a connection has been started.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	<p>From the operating system:</p> <pre>lsnrctl set connect_timeout time</pre> <p>From the LSNRCTL program:</p> <pre>lsnrctl> set connect_timeout time</pre>
Arguments:	time in seconds
Usage Notes:	None
Example:	<pre>lsnrctl> set connect_timeout 20 Connecting to (ADDRESS=(PROTOCOL=ipc)(KEY=iris)) LISTENER parameter "connect_timeout" set to 20 The command completed successfully</pre>

SET CURRENT_LISTENER

Purpose:	Sets or shows parameters for multiple listeners.
Prerequisites:	You must enter SET CURRENT_LISTENER from within the LSNRCTL utility
Password required if one has been set:	No
Syntax:	<p>From the operating system:</p> <pre>lsnrctl set current_listener [listener_name]</pre> <p>From the LSNRCTL program:</p> <pre>lsnrctl> set current_listener [listener_name]</pre>
Arguments:	listener name
Default Argument	LISTENER

SET CURRENT_LISTENER

Usage Notes:

If there is more than one listener on a node, any LSNRCTL command acts on the default listener (LISTENER) unless another listener has been set.

Any subsequent LSNRCTL commands within the same LSNRCTL session would then apply to the second listener, unless CURRENT_LISTENER were reset.

You can also display the current listener by using the LSNRCTL SHOW CURRENT_LISTENER command.

You must enter SET CURRENT_LISTENER from within the LSNRCTL utility. When you exit the utility, the setting will be lost.

Example:

```
lsnrctl> set current_listener [listener_name]
```

SET DISPLAYMODE

Purpose:	Changes the output of LSNRCTL SERVICE and LSNRCTL STATUS commands with additional information.
Password required if one has been set:	No
Syntax:	<p>From the operating system:</p> <pre>lsnrctl set displaymode <i>argument</i></pre> <p>From the LSNRCTL program:</p> <pre>lsnrctl> set displaymode <i>argument</i></pre>
Arguments:	<p>RAW (same as SET RAWMODE) COMPAT (for older versions) NORMAL VERBOSE (for support)</p> <p>Note: NORMAL and VERBOSE are not fully functional in this release</p>
Example:	<pre>lsnrctl> set displaymode compat lsnrctl> status Connecting to (ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC0)) STATUS of the LISTENER ----- Alias LISTENER Version TNSLSNR for Solaris: Version 8.1.5.0.0 Start Date 07-DEC-98 14:28:23 Uptime 0 days 0 hr. 30 min. 16 sec Trace Level off Security OFF SNMP OFF Listener Parameter File /private/dsteiner/installtest2/network/admin/listener.ora Listener Log File /private/dsteiner/installtest2/network/log/listener.log Services Summary... PLSExtProc has 1 service handler(s) sales has 3 service handler(s) The command completed successfully</pre>

SET LOG_DIRECTORY

Purpose:	Allows you to change the default directory where log files for the listener process are written.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	From the operating system: <code>lsnrctl set log_directory [directory]</code> From the LSNRCTL program: <code>lsnrctl> set log_directory [directory]</code>
Arguments:	directory path
Example:	<code>lsnrctl> set log_directory /usr/oracle/admin</code> Connecting to (ADDRESS=(PROTOCOL=ipc)(KEY=iris))LISTENER parameter "log_directory" set to /usr/oracle/admin The command completed successfully

SET LOG_FILE

Purpose:	Sets a non-default name for the log file.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	From the operating system: <code>lsnrctl set log_file [file_name]</code> From the LSNRCTL program: <code>lsnrctl> set log_file [file_name]</code>
Arguments:	file name
Example:	<code>lsnrctl> set log_file list.log</code> Connecting to (ADDRESS=(PROTOCOL=ipc)(KEY=iris))LISTENER parameter "log_file" set to list.log The command completed successfully

SET LOG_STATUS

Purpose:	Turns listener logging on or off.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	<p>From the operating system:</p> <pre>lsnrctl set set log_status [on off]</pre> <p>From the LSNRCTL program:</p> <pre>lsnrctl> set set log_status [on off]</pre>
Arguments:	ON OFF
Example:	<pre>lsnrctl> set log_status on</pre>

SET PASSWORD

Purpose:	Changes the password sent from the LSNRCTL utility to the listener process for authentication purposes only. To change the password on the listener itself, use the CHANGE_PASSWORD command.
Syntax:	<p>From the operating system:</p> <pre>lsnrctl set password</pre> <p>From the LSNRCTL program:</p> <pre>lsnrctl> set password</pre> <p>The syntax for unencrypted passwords is:</p> <pre>set password <i>password</i></pre>
Arguments:	password
Usage Notes:	<p>You may enter this command when you start up the shell or any time during your session. (You must enter the SET PASSWORD command before you can stop the listener.)</p> <p>The preferred, secure way to enter your password is in interactive mode. The listener supports encrypted and unencrypted passwords.</p>
Example:	<pre>lsnrctl> set password enter listener password: password</pre>

SET RAWMODE

Purpose: Changes the output of LSNRCTL SERVICE and LSNRCTL STATUS commands with additional information.

Password required if one has been set: No

Syntax: From the operating system:
`lsnrctl set rawmode argument`
 From the LSNRCTL program:
`lsnrctl> set rawmode argument`

Arguments: ON | OFF

Example:

```
lsnrctl> set rawmode on
lsnrctl> status
Connecting to (ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC))
STATUS of the LISTENER
-----
Alias                LISTENER
Version              TNSLSNR for Solaris: Version 8.1.5.0.0
Start Date           07-DEC-98 14:28:23
Uptime               0 days 0 hr. 9 min. 6 sec
Trace Level          off
Security             OFF
SNMP                 OFF
Listener Parameter File
/private/dsteiner/installtest2/network/admin/listener.ora
Listener Log File
/private/dsteiner/installtest2/network/log/listener.log
Services Summary
(SERVICE=(SERVICE_NAME=PLSExtProc)(INSTANCE=(INSTANCE_
NAME=PLSExtProc)(NUM=1)(INSTANCE_CLASS=ORACLE)))
(SERVICE=(SERVICE_NAME=dlsun17)(INSTANCE=(INSTANCE_
NAME=db1)(NUM=3)(INSTANCE_CLASS=ORACLE)))
The command completed successfully
```

SET SAVE_CONFIG_ON_STOP

Purpose:	Saves any changes made by the LSNRCTL SET command to the LISTENER.ORA if parameter is ON. The saving of all parameters occurs right before the listener exits, taking as much care as possible to preserve the formatting, comments, and letter case. To have all parameters saved right away, use the SAVE_CONFIG command.
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>lsnrctl set save_config_on_stop [on off]</code> From the LSNRCTL program: <code>lsnrctl> set save_config_on_stop [on off]</code>
Arguments:	ON OFF
Example:	<code>lsnrctl> set save_config_on_stop on</code>

SET STARTUP_WAITTIME

Purpose:	Sets the amount of time the listener sleeps before responding to a START command:
Prerequisites:	None
Password required if one has been set:	No
Syntax:	From the operating system: <code>lsnrctl set startup_waittime [time]</code> From the LSNRCTL program: <code>lsnrctl> set startup_waittime [time]</code>
Arguments:	time in seconds
Example:	<code>LSNRCTL> set startup_waittime 10</code> Connecting to (ADDRESS=(PROTOCOL=ipc)(KEY=iris))LISTENER parameter "startup_waittime" set to 10 The command completed successfully

SET TRC_DIRECTORY

Purpose:	Allows you to change the default location where trace files for the listener process will be written.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	From the operating system: <code>lsnrctl set trc_directory [directory]</code> From the LSNRCTL program: <code>lsnrctl> set trc_directory [directory]</code>
Arguments:	directory path
Example:	<code>lsnrctl> set trc_directory /usr/oracle/admin</code> Connecting to (ADDRESS=(PROTOCOL=ipc)(KEY=iris)) LISTENER parameter "trc_directory" set to /usr/oracle/admin The command completed successfully

SET TRC_FILE

Purpose:	Sets a non-default name for the trace file
Prerequisites:	None
Password required if one has been set:	No
Syntax:	From the operating system: <code>lsnrctl set trc_file [file_name]</code> From the LSNRCTL program: <code>lsnrctl> set trc_file [file_name]</code>
Arguments:	file name
Example:	<code>lsnrctl> set trc_file list.trc</code> Connecting to (ADDRESS=(PROTOCOL=ipc)(KEY=iris)) LISTENER parameter "trc_file" set to list.trc The command completed successfully

SET TRC_LEVEL

Purpose:	This command turns on tracing for the listener.
Prerequisites:	None
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>lsnrctl set trc_level level</code> From the LSNRCTL program: <code>lsnrctl> set trc_level level</code>
Arguments:	OFF USER ADMIN SUPPORT 0 4 10 16 <ul style="list-style-type: none"> ■ OFF 0 - No trace output ■ USER 4 - User trace information ■ ADMIN 10 - Administration trace information ■ SUPPORT 16 - WorldWide Customer Support trace information
Usage Notes:	Selecting USER provides a limited level of tracing; ADMIN provides a more detailed trace. This command overrides the setting in the LISTENER.ORA file. You must have set a valid password, if one is listed in the LISTENER.ORA file parameter <code>PASSWORDS_listener_name</code> to be able to use this command This command is identical to TRACE.
Example:	<code>lsnrctl> set trc_level admin</code>

SET_USE_PLUGANDPLAY

Purpose:	Instructs the listener to register its database information with an Oracle Names server.
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>lsnrctl set use_plugandplay [on off]</code> From the LSNRCTL program: <code>lsnrctl> set use_plugandplay [on off]</code>
Arguments:	ON OFF
Example:	<code>lsnrctl> set use_plugandplay on</code>

SHOW

Purpose: All of the SET commands listed except SET PASSWORD have equivalent SHOW commands. In response to one of the SHOW commands, LSNRCTL displays the current setting of the listener for that parameter.

Prerequisites: None

Password required if one has been set: see equivalent SET commands

Syntax: From the operating system:
`lsnrctl show [listener_name] [command]`

From the LSNRCTL program:

`lsnrctl> show [listener_name] [command]`

Arguments: listener name, show command

Usage Notes: The SHOW parameter can be shown, but not set, through LSNRCTL:

`show [listener_name] snmp_visible`

displays whether the listener is accessible to SNMP clients

Example: `lsnrctl> show`

The following operations are available after show

An asterisk (*) denotes a modifier or extended command:

```
connect_timeout
current_listener
displaymode
log_file
log_directory
log_status
password
raw_mode
save_config_on_stop
startup_waittime
trc_file
trc_directory
trc_level
use_plugandplay
```

SPAWN

Purpose:	Starts a program stored on the machine on which the listener runs, and which is listed with an alias in the LISTENER.ORA file.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	From the operating system: <code>lsnrctl spawn [<i>listener_name</i>] alias (arguments=arg1,arg2,)</code> From the LSNRCTL program: <code>lsnrctl> spawn [<i>listener_name</i>] alias (arguments=arg1,arg2,)</code>
Arguments:	<ul style="list-style-type: none"> ■ listener name ■ the alias of the program as listed in the listener configuration file ■ the arguments sent to the program that is to be spawned
Example:	<code>lsnrctl> spawn nstest_alias (ARGUMENTS='')</code>

START

Purpose:	Starts the named listener.
Prerequisites:	Listener must be stopped
Password required if one has been set:	No
Syntax:	From the operating system: <code>lsnrctl start [<i>listener_name</i>]</code> From the LSNRCTL program: <code>lsnrctl> start [<i>listener_name</i>]</code>
Arguments:	listener name If no listener name is entered, LISTENER is started by default.

START

Usage Notes: To start a listener configured in the LISTENER.ORA file with a name other than LISTENER, include that name.

For example, if the listener name is TCP_LSNR, enter:

```
lsnrctl start tcp_listener
```

Or, from the LSNRCTL program prompt, enter:

```
lsnrctl> start tcp_listener
```

Example:

```
lsnrctl> start
Starting /private/dsteiner/sales/bin/tnslsnr: please wait...

TNSLSNR for Solaris: Version 8.1.4.0.0 - Beta
System parameter file is
/private/dsteiner/sales/network/admin/listener.ora
Log messages written to
/private/dsteiner/sales/network/log/listener.log
Trace information written to
/private/dsteiner/sales/network/trace/listener.trc
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=dlsun1770)(PORT=1521))
(PROTOCOL_STACK=(PRESENTATION=TT)(SESSION=NS)))

Connecting to
(ADDRESS=(PROTOCOL=TCP)(HOST=dlsun1770)(PORT=1521))
STATUS of the LISTENER
-----
Alias                LISTENER
Version              TNSLSNR for Solaris: Version
8.1.4.0.0 - Beta
Start Date           08-JAN-99 15:50:10
Uptime               0 days 0 hr. 0 min. 0 sec
Trace Level          user
Security             OFF
SNMP                 OFF
Listener Parameter File
/private/dsteiner/sales/network/admin/listener.ora
Listener Log File
/private/dsteiner/sales/network/log/listener.log
Listener Trace File
/private/dsteiner/sales/network/trace/listener.trc
Services Summary...
  PLSExtProc          has 1 service handler(s)
The command completed successfully
```

STATUS

Purpose:	Displays basic information: version, start time, uptime, what LISTENER.ORA file is used, and whether tracing is turned on.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	From the operating system: <code>lsnrctl status [<i>listener_name</i>]</code> From the LSNRCTL program: <code>lsnrctl> status [<i>listener_name</i>]</code>
Arguments:	listener name
Usage Notes:	The status command allows you to perform the following: <ul style="list-style-type: none">▪ check the current setting of the logging and tracing options.▪ the list of database SIDs available through this listener. These are defined in the SID mapping in LISTENER.ORA.▪ whether a password is encrypted in LISTENER.ORA. (If you encrypt the listener password you can have only one password.)▪ whether the network listener can respond to queries from an SNMP-based network management system▪ the address(es) the TNSLSNR is listening on The SET DISPLAY and SET RAWMODE commands change the format of the output.

STATUS

Example: lsnrctl> status
 STATUS of the LISTENER

 Alias LISTENER
 Version TNSLSNR for Solaris: Version 8.1.5.0.0
 Start Date 07-DEC-98 14:28:23
 Uptime 0 days 0 hr. 9 min. 6 sec
 Trace Level off
 Security OFF
 SNMP OFF
 Listener Parameter File
 /private/dsteiner/installtest2/network/admin/listener.ora
 Listener Log File
 /private/dsteiner/installtest2/network/log/listener.log
 Services Summary...
 PLSExtProc has 1 service handler(s)
 db1 has 3 service handler(s)
 The command completed successfully

STOP

Purpose:	Stops the named listener.
Prerequisites:	The listener must be running
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>lsnrctl stop [listener_name]</code> From the LSNRCTL program: <code>lsnrctl> stop [listener_name]</code>
Arguments:	listener name
Usage Notes:	If you have configured passwords, you must use the SET PASSWORD command before you can use the STOP command. You must set the password from within the LSNRCTL program; you cannot set it from the operating system command line. The method for setting the password depends on whether you are using the encrypted password feature. If you are <i>not</i> using an encrypted password, enter the password on the LSNRCTL command line. Be careful when stopping a listener. On some platforms and with some protocols, when a listener is stopped any Net8 connections currently running are shut down. In some situations the connections continue, but it is then not possible to start the listener again until the running processes have been closed. It is good practice to send a warning message to all network users before stopping a listener.
Example:	<code>lsnrctl> stop</code>

TRACE

Purpose:	Turns on tracing for the listener.
Prerequisites:	valid password required
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>lsnrctl trace off user admin support [<i>listener_name</i>]</code> From the LSNRCTL program: <code>lsnrctl> trace off user admin support [<i>listener_name</i>]</code>
Arguments:	[OFF USER ADMIN SUPPORT] [<i>listener name</i>] <ul style="list-style-type: none"> ■ OFF- No trace output ■ USER - User trace information ■ ADMIN - Administration trace information ■ SUPPORT - WorldWide Customer Support trace information
Usage Notes:	USER provides a limited level of tracing. ADMIN provides a more detailed trace. This command overrides the setting in the LISTENER.ORA file. This command has the same functionality as SET TRC_LEVEL.
Example:	<code>lsnrctl> trace admin listener</code>

VERSION

Purpose:	Displays the current TNS listener, and protocol version.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	From the operating system: <code>lsnrctl version [<i>listener_name</i>]</code> From the LSNRCTL program: <code>lsnrctl> version [<i>listener_name</i>]</code>
Arguments:	<i>listener name</i>
Example:	<code>lsnrctl> version listener1</code>

Oracle Names Control Utility (NAMESCTL)

The Oracle Names Control Utility (NAMESCTL) is a tool that you run from the operating system prompt to start and control Oracle Names servers. It contains several types of commands:

- Operational commands such as START, STOP, RESTART, and so forth.
- Modifier commands, such as SET *property*
- Informational commands, such as STATUS, SHOW *property*, and PING
- Command utility operational commands, such as EXIT, QUIT, and HELP

You can use the NAMESCTL control utility to perform basic management functions on one or more Oracle Names servers. Additionally, you can view and change Oracle Names server parameter settings such as RESET_STATS_INTERVAL and TRACE_LEVEL.

Parameter Options

When loading NAMESCTL, any valid parameter settings can be passed to the program to override the default or configured settings. For example:

```
namesctl namesctl.trace_level=admin
```

would load NAMESCTL and turn on tracing to the ADMIN level, regardless of the currently configured value of NAMESCTL.TRACE_LEVEL.

SET and SHOW Modifiers

You can use the modifier SET to change some parameter values for an Oracle Names server or the NAMESCTL environment during the NAMESCTL control utility session. For example, the following sequence sets the node to control and changes its trace level.

```
namesctl> set server dolphin.com  
namesctl> set trace_level admin
```

The first modifier sets the node to DOLPHIN.COM. Subsequent commands are directed to DOLPHIN.COM. The second modifier sets the server DOLPHIN.COM's trace level. The server will then begin tracing at the ADMIN level.

NAMESCTL's Distributed Operation

The Oracle Names Control Utility operates on an Oracle Names server on the same machine as any other Oracle Names servers in the network. This is very useful when a single administrator is managing all of the Oracle Names servers in a region, or wants to check the availability of a specific Oracle Names server.

Most commands accept the name of an Oracle Names server as the last argument indicating which Oracle Names server to perform the command against. If omitted, the current SET Oracle Names server is used. For example:

```
show system_queries dolphin.acme
```

will display the system queries on the Oracle Names server DOLPHIN.ACME and when they will next occur. Prior to performing a series of commands against an individual Oracle Names server, enter

```
namesctl> set server onames_server
```

Security

If the NAMES.PASSWORD parameter is set in the NAMES.ORA, the NAMESCTL control utility will require a SET PASSWORD command for any sensitive operations, such as starting or stopping the Oracle Names server.

If NAMESCTL.SERVER_PASSWORD parameter is set in the SQLNET.ORA file on the node running NAMESCTL, you will *not* be prompted to use the SET PASSWORD command each time a secure operation is performed.

If you are concerned with the security implications of explicitly putting an Oracle Names server password in the administrator's client SQLNET.ORA file, you can omit the NAMESCTL.SERVER_PASSWORD parameter and always use the command:

```
namesctl> set password
```

When passed over the network, the password is encrypted regardless of how it set in the NAMES.ORA file unless the NAMESCTL.INTERNAL_ENCRYPT_PASSWORD is set to FALSE in the SQLNET.ORA file. If this parameter is set to FALSE, the password is not encrypted.

Confirmation Mode in NAMESCTL

Some of the NAMESCTL commands require your confirmation before they are executed. When you issue the command, you are prompted:

```
confirm:[yes or no]
```

Enter "yes" to execute the command; enter "no" to cancel the command.

You can turn confirmation mode off by using by setting the parameter NAMESCTL.NOCONFIRM = TRUE in SQLNET.ORA. Note that with this parameter set to OFF, all commands execute without asking for confirmation.

NAMESCTL Commands

The following commands are available through the Oracle Names Control Utility (NAMESCTL):

DELEGATE_DOMAIN

Purpose:	Defines a domain as the start of a subregion of the current region
Prerequisites:	none
Password required if one has been set:	No
Syntax:	<p>From the operating system prompt:</p> <pre>namesctl delegate_domain domain_name onames_server [description=(address_list=](address= (protocol_address_ information))[])]</pre> <p>From the NAMESCTL program:</p> <pre>delegate_domain domain_name onames_server [(description=address_list=] (address= (protocol_ address_information))[])]</pre>
Arguments:	<i>domain_name</i> , <i>onames_server</i> and the address are mandatory. <i>domain_name</i> and <i>onames_server</i> must be legal domain names. The address must be a legal TNS address.
Usage Notes:	<p>This command provides a dynamic way to subdivide the namespace.</p> <p>The domain <i>domain_name</i> is defined with <i>onames_server</i> as the name of the Oracle Names server. <i>onames_server</i> is defined with its address.</p> <p>Unless a domain is delegated from a region, the Oracle Names servers in that region will assume authority over all sub-domains. In order to delegate a domain, you must first create a new region.</p> <p>Once a domain is delegated, the Oracle Names servers in the current region will forward subsequent operations to the sub-region where the domain is administered by Oracle Names servers.</p>
Examples:	<pre>namesctl> delegate_domain webwidgets.acme.com ns1.webwidgets.acme.com (address=(protocol=tcp) (host=fred.webwidgets.acme.com) (port=1575))</pre>

DOMAIN_HINT

Purpose:	If the Oracle Names server is not in the root region, identifies the location of an Oracle Names server in the root region.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	<p>From the operating system:</p> <pre>namesctl domain_hint domain_name onames_server [description=(address_list=(address= (protocol_address_ information))))</pre> <p>From the NAMESCTL program:</p> <pre>domain_hint domain_name onames_server [description=(address_list=(address= (protocol_address_ information))))</pre>
Arguments:	<i>domain_name</i> , <i>onames_server</i> and the address are mandatory. <i>domain_name</i> and <i>onames_server</i> must be legal domain names. The address must be a legal TNS address.
Usage Notes:	<p>This command provides a dynamic way to define the path to other regions in the namespace.</p> <p>The domain <i>domain_name</i> is defined with <i>onames_server</i> as the name of its Oracle Names server. <i>onames_server</i> is defined with its address.</p> <p>Any region that is not the root region will need at least the root region defined using this command in order to find objects in any other region. You may provide additional hints as optimizations to provide local Oracle Names servers with direct access to certain other regions.</p>
Examples:	<pre>namesctl> domain_hint acme.com ns0.acme.com (address=(protocol=tcp) (host=top.acme.com) (port=1575))</pre>

DUMP_TNSNAMES

Purpose:	Writes the names and address of all address defined in the local region into the TNSNAMES.ORA file.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	From the operating system: <code>namesctl dump_tnsnames</code> From the NAMESCTL program: <code>dump_tnsnames</code>
Arguments:	None
Usage Notes:	Writes the name and address of all addresses defined in the local region into TNSNAMES.ORA (everything with an address-type record (A.SMD)). The command will create or partially overwrite the TNSNAMES.ORA file. Any entries in the existing TNSNAMES.ORA which are not defined in the Oracle Names servers will remain. Any definitions in TNSNAMES.ORA which are also defined in the Oracle Names servers will be overwritten. And entries which are defined in the Oracle Names servers but not in the TNSNAMES.ORA will be added.
Examples:	<code>namesctl> dump_tnsnames</code>

EXIT

Purpose:	The EXIT command closes the NAMESCTL program.
Prerequisites:	The NAMESCTL program must be loaded.
Password required if one has been set:	No
Syntax:	From the NAMESCTL program: EXIT
Arguments:	None
Usage Notes:	EXIT has no effect on any Oracle Names servers. It affects only the NAMESCTL program. The EXIT command is identical to the QUIT command.
Example:	<pre>namsctl> exit namesctl finished.</pre>

FLUSH

Purpose:	Drops all stored non-authoritative data from the Oracle Names server cache.
Prerequisites:	Only relevant with an environment with multiple regions. (In central administration there is no non-authoritative data.)
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system prompt: <code>namesctl flush [onames_server] ...</code> From the NAMESCTL program: <code>flush [onames_server] ...</code>
Arguments:	Zero or more Oracle Names servers separated by a space. When no arguments are supplied, only the current Oracle Names server's cache is flushed of the foreign names
Usage Notes:	FLUSH erases <i>all foreign data</i> that has been cached. Typically, you should flush the foreign data cache when: <ul style="list-style-type: none">- A large volume of data changes in the network and the normal TTL aging mechanism will take too long.- When unidentifiable errors in name resolution of cached foreign data are occurring. Flushing all foreign data from the cache forces it to be looked up again when it is requested the next time.
Examples:	<code>namesctl>flush</code> <code>Confirm [yes or no]: yes</code>

FLUSH_NAME

Purpose:	Drops one or more specific non-authoritative names from the current Oracle Names server's cache.
Prerequisites:	Only meaningful with an environment with multiple regions. (In central administration, there is no non-authoritative data.)
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system prompt: <code>namesctl flush_name name</code> From the NAMESCTL program: <code>flush_name name</code>
Arguments:	A single name
Usage Notes:	FLUSH_NAME erases only data cached from outside the Oracle Names server's region (that is, non-authoritative data). It is typically flushed when a name is behaving unusually, suggesting the source copy may have changed. FLUSH_NAME removes the name from the current foreign data cache as well as any other Servers between the current region and the authoritative region. Names are flushed from the current Oracle Names server. The current Oracle Names server is either the default preferred Oracle Names server or the one set by using the SET SERVER command.
Examples:	<code>namesctl>flush_name mountain.acme.com</code>

HELP

Purpose:	Provides details of the NAMESCTL commands.
Prerequisites:	None.
Password required if one has been set:	No
Syntax:	From the operating system prompt: <code>namesctl help [command]</code> From the NAMESCTL program: <code>help [command]</code>
Arguments:	<code>help command</code>
Usage Notes:	Help provides brief reminders of the function of each command in NAMESCTL. When no arguments are supplied, help shows the list of valid commands. When you supply an argument, a one line description of that command's function is displayed.
Example:	<code>namesctl> help</code> The following operations are available: An asterisk (*) denotes a modifier or extended command: <code>exit</code> <code>flush</code> <code>flush_name</code> <code>log_stats</code> <code>ping</code> <code>query</code> <code>quitreload</code> <code>repeat*</code> <code>reset_stats</code> <code>restart</code> <code>set*</code> <code>show</code> <code>shutdown</code> <code>start</code> <code>startup</code> <code>status</code> <code>stop</code> <code>version</code>

LOAD_TNSNAMES

Purpose:	Loads all net service names and addresses defined in one or more TNSNAMES.ORA files into an Oracle Names server.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	<p>From the operating system prompt:</p> <pre>namesctl load_tnsnames <i>file_name</i> [<i>file_name</i>] ...</pre> <p>From the NAMESCTL program:</p> <pre>load_tnsnames <i>file_name</i> [<i>file_name</i>] ...</pre>
Arguments:	One or more file names containing ' <i>name</i> = <i>address</i> ' definitions in a TNSNAMES.ORA format.
Usage Notes:	This command would typically be used once when a site begins using Oracle Names after having used TNSNAMES.ORA files. It is an alternative to upgrading the database tables which were maintained by Oracle Network Manager. Run this command once per region. The names defined during this operation will be defined permanently and will be propagated from one server to all the rest in the region.
Example:	<pre>namesctl>load_tnsnames network/admin/tnsnames.ora Name: koala.lab.npd.us.oracle.com Response status: normal, successful completion Name: devdd.rdbms.us.oracle.com Response status: normal, successful completion Name: envyd.lab.npd.us.oracle.com Response status: normal, successful completion Name: stealth.npd.us.oracle.com Response status: normal, successful completion Name: null.us.oracle.com Response status: normal, successful completion Name: slime.lab.npd.us.oracle.com Response status: normal, successful completion Name: felix.hp.us.oracle.com Response status: normal, successful completion Name: dtnet1.dec.oracle.com Response status: normal, successful completion Name: devds.rdbms.us.oracle.com Response status: normal, successful completion</pre>

LOG_STATS

Purpose:	Logs the current set of statistics to the configured log file for that Oracle Names server.
Prerequisites:	None
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system prompt: <code>namesctl log_stats [onames_server] ...</code> From the NAMESCTL program: <code>log_stats [onames_server] ...</code>
Arguments:	Zero or more Oracle Names server names separated by a space. When no arguments are supplied, only the statistics for the current Oracle Names server are reset.
Usage Notes:	Statistics may be logged if the STATUS command or other behavior indicates some data that you would like to capture in the log. LOG_STATS does not affect the current LOG_STATS_INTERVAL.
Example:	<code>namesctl> log_stats</code> Statistics counters logged.

PASSWORD

Purpose:	Registers the password for privileged Oracle Names server operations, such as RELOAD and STOP.
Prerequisites:	The NAMESCTL program must be loaded.
Password required if one has been set:	N/A
Syntax:	From the NAMESCTL program: <code>password [<i>password</i>]</code>
Arguments:	Text string matching the value encrypted in the <code>NAMES.PASSWORD</code> parameter in NAMES.ORA.
Usage Notes:	<p>PASSWORD does not change the Oracle Names server's password. It simply sets a NAMESCTL variable. Then, the value stored is sent from NAMESCTL with any command request to the Oracle Names server, and the value is compared to the value configured on the Oracle Names server. If they match, operations requiring passwords are allowed.</p> <p>Only "privileged" operations are affected, that is, operations that alter the functioning of the Oracle Names server. Operations such as SHOW or STATUS are not considered privileged, and do not require a password.</p> <p>The password can either be passed as an argument of the PASSWORD command, or if no argument is given, it will be prompted for. Note that the input is not displayed on the screen as it is typed.</p> <p>When passed over the network the password is ALWAYS encrypted, regardless of how it is set.</p>
Example:	<pre>namesctl> password open_sesame namesctl> password Enter name server password:</pre>

PING

Purpose:	Contacts the current Oracle Names server, or named server(s), and display the request/response time.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	From the operating system prompt: <code>namesctl ping [onames_server] ...</code> From the NAMESCTL program: <code>ping [onames_server] ...</code>
Arguments:	Zero or more Oracle Names server names separated by a space. When no arguments are supplied, only the current Oracle Names server is pinged.
Usage Notes:	Ping ensures that an Oracle Names server is functioning and shows typical response times from the location of the NAMESCTL user to an Oracle Names server.
Example:	<code>namesctl> ping nserver.com</code> Round trip time is 0.04 seconds

QUERY

Purpose: Tests or retrieves the contents of a network object stored in the Oracle Names server.

Prerequisites: None

Password required if one has been set: No

Syntax From the operating system prompt:
`namesctl query object_name [object_type] [modifiers]`
 From the NAMESCTL program:
`query object_name [object_type] [modifiers]`

COMMON OBJECT TYPES:

- A.SMD—Network addresses, as with database service definitions or net service names
- CNAME.SMD—Aliases (sometimes referred to as "canonical name")
- DL.RDBMS.OMD—Global database links
- DLCR.RDBMS.OMD—Global database link qualifiers
- NS.SMD—Oracle Names server addresses. System data used to communicate between Oracle Names servers.
- V1ADD.NPO.OMD—SQL*Net Version 1 connect string

VALID MODIFIERS:

- AUTHORITY—Forces the query to be resolved at the source of the data (in the administrative region where the data is considered local) even if the data is in the local cache. This could be used if the administrator suspects that the data has changed at the source.
- NOFORWARD—Query for the data, but don't forward the request. When the data is not local, and noforward is specified, the query will not be resolved.
- TRACE—Allows a trace of the path to the answer. This is useful whenever you want to find out which Oracle Names servers the request went to.

Arguments: Mandatory network object name and network object type

QUERY

Usage Notes:

QUERY can be used to test that a defined piece of data can be found, and that the contents are correct.

The QUERY command always operates on the current Oracle Names server, either the default, or as specified using the SET SERVER command.

If the QUERY command is used with just a name as a parameter, the Oracle Names server responds with the number of pieces of data with that name, and the time required to complete the operation.

If the QUERY command is used with the name and type supplied as arguments; the specific name is looked up and returned to the user.

The QUERY command can take multiple arguments if appropriate. For example:

```
query sales.com a.smd authority trace
```

Example:

```
namesctl> query bones.dem.medicine a.smd
Total response time:0.04 seconds
Response status:normal, successful completion
Authoritative answer:yes
Number of answers:1
Canonical name:bones.dem.medicine
TTL: 1 day
Alias translations:
  from:bones.dem.medicine
  to: bones.dem.medicine
Answers:
  data type is "a.smd"
  Syntax is ADDR:... (DESCRIPTION=(ADDRESS=
(PROTOCOL=TCP) (Host=bones-pc)
(Port=1521)) (CONNECT_DATA=(SERVICE_NAME=bones.dem.medicine)))
```

QUIT

Purpose:	Quits the NAMESCTL program.
Prerequisites:	The NAMESCTL program must be loaded.
Password required if one has been set:	No
Syntax:	From the NAMESCTL program: QUIT
Arguments:	None
Usage Notes:	QUIT has no effect on any Oracle Names servers. It affects only the NAMESCTL program. The QUIT command is functionally equivalent to the EXIT command.
Example:	namesctl> quit nl-00851: namesctl finished

REGISTER

Purpose:	Registers a network object to an Oracle Names server
Password required if one has been set:	No
Syntax:	From the NAMESCTL program: register <i>object_name</i> [-t <i>service_type</i>] [-d description=(address_list=(address= (<i>protocol_address_information</i>)))] [-h <i>onames_server</i>] [-l <i>listener_name</i>] TYPE OF SERVICE: <ul style="list-style-type: none"> ■ ORACLE_DATABASE ■ ORACLE_LISTENER ■ ORACLE_NAMESERVER
Arguments:	Mandatory object name. The service, address data, and host are not necessary to make the registration process appear to work. However, they are necessary to make the registration useful. In other words, an object name registered without an address cannot be used.

REGISTER

Usage Notes: Provides a manual mechanism for registering a service, its type, its hostname, and its address. Both the type of service and the data may be any valid string, but the typical registration has either "database" or "listener" as type of service, and the TNS address as the data.

The object registration is propagated to all other well known Oracle Names servers in the region.

Example:

```
namesctl> register parts -t oracle_database -d (DESCRIPTION=
(AADDRESS= (PROTOCOL=TCP)(HOST=nineva)(PORT=1575)) (CONNECT_
DATA=(SERVICE_NAME=db3)))
```

RELOAD

Purpose: Forces the server to check immediately for data changes in its administrative region, and if there are any, reloads all database service names, net service names, global database links, and aliases.

Prerequisites: None

Password required if one has been set: Yes
If a password is set, the SET PASSWORD command must be issued prior to this command.

Syntax: From the operating system prompt:

```
namesctl reload [onames_server] ...
```

From the NAMESCTL program:

```
reload [onames_server] ...
```

Arguments: Zero or more Oracle Names server names separated by a space. When no arguments are supplied, only the current Oracle Names server is reloaded.

Usage Notes: All Oracle Names servers load their data directly from the database specified by the NAMES.ADMIN_REGION configuration parameter in the NAMES.ORA file.

In an environment with multiple regions, RELOAD affects only the data for the current administrative region. All foreign data in the cache is unchanged.

Example:

```
namesctl> reload
Server reloaded.
```

REORDER_NS

Purpose:	Creates the file which lists local Oracle Names servers and their listening addresses.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	<p>From the operating system prompt:</p> <pre>namesctl reorder_ns [description=(address_list= (address= (protocol_address_information)))]</pre> <p>From the NAMESCTL program:</p> <pre>reorder_ns [description=(address_list= (address= (protocol_address_information)))]</pre>
Arguments:	An optional Oracle Names server address will be used as the initial server to contact.
Usage Notes:	<p>This command generates the file which defines Oracle Names server names and addresses to enable clients to contact Oracle Names servers for name lookup.</p> <p>The REORDER_NS command does the following:</p> <ol style="list-style-type: none"> 1. Finds an Oracle Names server: 1. It looks in the SQLNET.ORA for a preferred Oracle Names server parameter, 2. tries calling a well-known servers, tries locating a local Oracle Names server using TCP/IP on port 1575. It uses the address argument if it is present. 2. Sends a query for all the Oracle Names servers in the local region. 3. Sends a PING to each of these servers. 4. Sorts the list of servers by increasing order of response time. 5. Writes an Oracle Names server List with the sorted list of names and addresses.
Example:	<pre>namesctl> reorder_ns (address= (protocol=tcp) (host=nineva) (port=1575))</pre>

REPEAT

Purpose: Used to perform QUERY, REGISTER, TIMED_QUERY, or UNREGISTER multiple times to compute average return rates.

Prerequisites: None

Password required if one has been set: No

Syntax: From the operating system prompt:
`namesctl repeat number query type`
From the NAMESCTL program:
`repeat number query type`
where *number* is an integer and *type* is as shown in the QUERY command.

Arguments: None

Usage Notes: Repeat is useful for understanding the average response time over a number of requests.

Do not specify too large a number here; while the number of iterations are occurring, the NAMESCTL program cannot do anything else.

Example:

```
namesctl> repeat 10 query manatee a.smd
Number of requests: 10
Average response time: 0.01 seconds
Minimum response time: 0.01 seconds
Maximum response time:0.04 seconds
Total response time:0.14 seconds
Response status:normal, successful completion
Authoritative answer:yes
Number of answers: 1
TTL: 1 day
Answers:
    data type is "a.smd"
    Syntax is ADDR:(DESCRIPTION=(ADDRESS=
(PROTOCOL=TCP)(Host=salmon)
(Port=1522))(CONNECT_DATA=(SID=otter)))
```

RESET_STATS

Purpose:	Resets the Oracle Names server statistics to the original values of the Oracle Names server at startup.
Prerequisites:	None
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system prompt: <code>namesctl reset_stats [onames_server] ...</code> From the NAMESCTL program: <code>reset_stats [onames_server] ...</code>
Arguments:	Zero or more Oracle Names server names separated by a space. When no arguments are supplied, only the current Oracle Names server's statistics are reset.
Usage Notes:	RESET_STATS has the same effect as waiting for the RESET_STATS_INTERVAL to conclude, except that it happens immediately.
Example:	<code>namesctl> reset_stats</code> Confirm [yes or no]: yes Server statistics reset.

RESTART

Purpose:	Initiates a reset of an Oracle Names server to its original state at startup.
Prerequisites:	None
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system prompt: <code>namesctl restart [onames_server] ...</code> From the NAMESCTL program: <code>restart [onames_server] ...</code>
Arguments:	Zero or more Oracle Names server names separated by a space. When no arguments are supplied, only the current Oracle Names server is restarted.
Usage Notes:	RESTART is the same as STARTUP except that the Oracle Names server is already running. Data is reloaded, statistics are reset, and all foreign data is flushed. Valid foreign cache data (that is, data with a TTL greater than 0) is retrieved from the checkpoint files. (The TTL value must be set to more than 0.)
Example:	<pre>namesctl> restart Confirm [yes or no]: yes Server restarted.</pre>

SET

Purpose:	Lists the parameter values that can be set using the SET command.
Prerequisites:	None
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	<code>namesctl> set</code>
Arguments:	None
Usage Notes:	<p>You must have set a valid password to be able to use this command if one is listed in the listener configuration parameter, <code>PASSWORDS_listener_name</code>.</p> <p>If there are any passwords in the listener configuration file, you must use the SET PASSWORD command before you can use the SET command.</p> <p>You must set the password from within the LSNRCTL program; you cannot set it from the operating system command line. The method for setting the password depends on whether you are using the encrypted password feature. If you are <i>not</i> using an encrypted password, enter the password on the LSNRCTL command line.</p>
Example:	<pre>namesctl> set</pre> <p>The following operations are available after set</p> <p>An asterisk (*) denotes a modifier or extended command:</p> <pre>cache_checkpoint_interval default_domain forwarding_available log_file_name log_stats_interval namesctl_trace_level password requests_enabled reset_stats_interval save_config_interval save_config_on_stop server trace_file_name trace_level</pre>

SET CACHE_CHECKPOINT_INTERVAL

Purpose: Sets how often all collected information about foreign regions is saved in the Oracle Names server cache file.

Password required if one has been set: No

Syntax: From the operating system:
`namesctl set cache_checkpoint_interval time`
From NAMESCTL program:
`set cache_checkpoint_interval time`

Arguments: Time is in *seconds*
For example, to increase the CACHE_CHECKPOINT_INTERVAL to 36 hours, the following will work:
`set cache_checkpoint_interval 129600`

Usage Notes Minimum: 10 seconds
Maximum: 259200 (3 days)
Default: 0 (disabled)

Example: `namesctl> set cache_checkpoint_interval 10`

SET DEFAULT_DOMAIN

Purpose:	Sets or changes the default domain for the NAMESCTL client.
Prerequisites:	The NAMESCTL program must be loaded.
Password required if one has been set:	No
Syntax:	From the NAMESCTL program: <pre>set default_domain domain_name</pre>
Arguments:	one domain name
Usage Notes:	<p>The existence of the DEFAULT_DOMAIN parameter allows names to be unqualified for names in that domain. For example, with DEFAULT_DOMAIN set to US.ACME the global name WIDE.US.ACME could be queried using:</p> <pre>namesctl> query wide</pre> <p>The initial value of DEFAULT_DOMAIN is set when the NAMESCTL program is started from the NAMES.DEFAULT_DOMAIN parameter in the SQLNET.ORA.</p> <p>When no arguments are specified, the default is read and assigned from the SQLNET.ORA.</p> <p>SET DEFAULT_DOMAIN could be used to simplify working on a set of names within a single domain, then a set in another.</p>
Example:	<pre>namesctl> set default_domain us.acme Default domain is now "US.ACME"</pre>

SET FORWARDING_AVAILABLE

Purpose:	Turns on or off name request forwarding for an Oracle Names server
Prerequisites:	Oracle Names server must be running
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>namesctl set forwarding_available on off yes no</code> From the NAMESCTL program: <code>set forwarding_available off</code>
Arguments:	ON OFF YES NO
Default Value:	OFF
Usage Notes:	<p>This setting is intended only for Oracle Names servers that have no local clients and are exclusively handling requests from foreign Oracle Names servers. This usually would only apply to Oracle Names servers in the root region when the root is configured without clients or services. If such a server is a performance bottleneck in cross-region request processing then disabling forwarding in that Oracle Names server will cut its workload in half. Rather than forward the request and return the answer the Oracle Names server simply tells the requestor the address of the Oracle Names server that can answer the request. Note that there is no overall reduction in work; the work is simply displaced from the non-forwarding Oracle Names server to the requesting Oracle Names server.</p> <p>WARNING: If SET FORWARDING_AVAILABLE is set to off, any clients who rely directly on that Oracle Names server will be unable to resolve foreign names. Clients are not capable of redirecting their requests as Oracle Names servers would. Their requests will fail at that point, even if other Oracle Names servers are listed in the NAMES.PREFERRED_SERVERS configuration parameter.</p>
Example:	<code>namesctl> set forwarding_available off</code> Request processing is now disabled.

SET LOG_FILE_NAME

Purpose:	Changes the log file name. The default name is NAMES.LOG.
Prerequisites:	None
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>namesctl set log_file_name file_name</code> From the NAMESCTL program: <code>set log_file_name file_name</code>
Arguments:	file name of the log file.
Usage Notes:	The LOG_FILE_NAME changes the destination of all logging messages.
Example:	<code>namesctl> set log_file_name namesvr1</code>

SET LOG_STATS_INTERVAL

Purpose:	Changes the frequency with which the statistics are logged to the log file.
Prerequisites:	None
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>namesctl set log_stats_interval time</code> From the NAMESCTL program: <code>namesctl> set log_stats_interval time</code>
Arguments:	Time is in seconds or [<i>n</i> > DAY[S]] [<i>hh</i> >: <i>mi</i> >: <i>ss</i> >] For example, to increase the LOG_STATS_INTERVAL to 36 hours, either of the following will work: <code>set log_stats_interval 129600</code> <code>set log_stats_interval 1 day 12:00:00</code> You can specify any valid combination, such as the number of days combined with number of hours, minutes, and seconds; or just the number in hours.
Restrictions:	Minimum Value: 10 seconds Maximum Value: no maximum Special Value: 0 (which means never reset) Default value: 0 (no logging)
Usage Notes:	The LOG_STATS_INTERVAL value is initially set based on the value configured in the Oracle Network Manager, or the value in NAMES.LOG_STATS_INTERVAL in the SQLNET.ORA file when the Oracle Names server is loaded. By default, the value is 0 (no logging). This command is intended to override that value during server operation.
Example:	<code>namesctl> set log_stats_interval 7200</code> Statistic counter logging interval is now 2 hours

SET NAMESCTL_TRACE_LEVEL

Purpose:	Sets the level at which the NAMESCTL program can be traced.
Prerequisites:	None
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>namesctl set namesctl_trace_level off user admin support</code> From the NAMESCTL program: <code>namesctl> set namesctl_trace_level off user admin support</code>
Arguments:	OFF USER ADMIN SUPPORT
Usage Notes:	Tracing assists in diagnosing unexpected or unidentifiable failures in processing the NAMESCTL program. Tracing writes a series of events from normal NAMESCTL processing to an operating system file for review by the administrator. Tracing output is at three levels OFF (none), USER (basic information), or ADMIN. When no arguments are supplied, the setting is reset to the value in the client's SQLNET.ORA file. The default setting is OFF.
Example:	<code>namesctl> set namesctl_trace_level admin</code> Controller's local trace level changed from 0 to 4

SET PASSWORD

Purpose:	Register the password for privileged Oracle Names server operations such as RELOAD and STOP.
Prerequisites:	The NAMESCTL program must be loaded.
Password required if one has been set:	N/A
Syntax:	From the NAMESCTL program: <pre>namesctl> set password [<i>password</i>]</pre>
Arguments:	Text string matching the value stored in the current Oracle Names server parameter NAMES.PASSWORD.
Usage Notes:	<p>SET PASSWORD does not change the Oracle Names server's password. It simply sets a NAMESCTL variable that is sent over to the Oracle Names server with any NAMESCTL command and is compared to the value configured on the Oracle Names server. If they match, operations requiring passwords are allowed.</p> <p>Only "privileged" operations are affected, that is, operations that alter the functioning of the Oracle Names server. Operations such as SHOW or STATUS are not considered privileged, and do not require a password.</p> <p>The password can either be passed as an argument of the SET PASSWORD command, or if no argument is given, it will be prompted for. Note that the input is not displayed on the screen as it is typed.</p> <p>When passed over the network the password is ALWAYS encrypted, regardless of how it is set.</p>
Example:	<pre>namesctl> set password open_sesame namesctl> set password enter name server password:</pre>

SET REQUESTS_ENABLED

Purpose:	Determine whether the current Oracle Names server will respond to requests.
Prerequisites:	None
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>namesctl requests_enabled on off</code> From the NAMESCTL program: <code>namesctl> set requests_enabled on off</code>
Arguments:	ON OFF
Usage Notes:	Setting this property to OFF will send refusals to all clients that approach with Names requests. This is primarily useful for diagnostics when an Oracle Names server is functioning unexpectedly.
Example:	<code>namesctl> set requests_enabled off</code> Confirm [yes or no]: yes General request processing is now disabled

SET RESET_STATS_INTERVAL

Purpose:	Changes the time between the statistics being reset to zero or initial values in the current server.
Prerequisites:	None
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>namesctl set reset_stats_interval <i>time</i></code> From the NAMESCTL program: <code>namesctl> set reset_stats_interval <i>time</i></code>
Arguments:	Time is in one of the following formats: seconds <code>[<i>n</i> DAY[S]] [<i>hh:mi:ss</i>]</code> For example, to increase the RESET_STATS_INTERVAL to 72 hours, either of the following will work: <code>set reset_stats_interval 259200</code> <code>set reset_stats_interval 3 days</code>
Restrictions:	Minimum Value: 10 seconds Maximum Value: no maximum Default value: 0 (never reset)
Usage Notes:	The RESET_STATS_INTERVAL value is initially set based on the NAMES.RESET_STATS_INTERVAL parameter when the Oracle Names server is loaded. This command is intended to override that value during Oracle Names server operation.
Example:	<code>namesctl> set reset_stats_interval 1 day</code> Statistic counter reset interval is now 24 hours

SET SAVE_CONFIG_INTERVAL

Purpose:	Saves any changes made by the NAMESCTL SET command to the NAMES.ORA at an interval.
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>namesctl set save_config_interval time</code> From NAMESCTL program: <code>set save_config_interval time</code>
Arguments:	Time is in seconds
Example:	<code>namesctl> set save_config_interval 10</code>

SET SAVE_CONFIG_ON_STOP

Purpose:	Saves any changes made by the NAMESCTL SET command to the NAMES.ORA if parameter is ON. The saving of all parameters occurs right before NAMESCTL exits, taking as much care as possible to preserve the formatting, comments, and letter case.
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>namesctl set save_config_on_stop on off</code> From the NAMESCTL program: <code>namesctl> set save_config_on_stop on off</code>
Arguments:	ON OFF
Example:	<code>namesctl> set save_config_on_stop on</code>

SET SERVER

Purpose:	Change the current Oracle Names server.
Prerequisites:	The NAMESCTL program must be loaded
Password required if one has been set:	No
Syntax:	From the NAMESCTL program: <pre>namesctl> set server [onames_server] [description=(address_list= (address= (protocol_address_ information)))]</pre>
Arguments:	valid Oracle Names server or valid Oracle Names server address
Usage Notes:	<p>SET SERVER allows switching between multiple Oracle Names servers while running the NAMESCTL utility. The qualifier can be a name where the name is defined in the memory of the current Oracle Names server, or it can be the TNS address of any Oracle Names server.</p> <p>The Oracle Names server name specified is resolved through normal name lookup. Another Oracle Names server can only be set if the current Oracle Names server knows or can retrieve its address. If no current Oracle Names server is set, you must type a TNS address to complete this command. If there are no arguments, use NAMES.PREFERRED_SERVERS.</p>
Example:	<pre>namesctl> set server server1.us.acme</pre>

SET TRACE_FILE_NAME

Purpose: Changes the trace destination file name

Prerequisites: None

Password required if one has been set: No

Syntax: From the operating system:
`namesctl set trace_file_name file_name`
From NAMESCTL program:
`namesctl> set trace_file_name file_name`

Arguments: file name

Example: `namesctl> set trace_file_name namesvr1`

SET TRACE_LEVEL

Purpose:	Changes the TRACE_LEVEL for tracing the current Oracle Names server.
Prerequisites:	None
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>namesctl set trace_level off user admin support</code> From the NAMESCTL program: <code>namesctl> set trace_level off user admin support</code>
Arguments:	OFF USER ADMIN SUPPORT <ul style="list-style-type: none">■ OFF- No trace output■ USER - User trace information■ ADMIN - Administration trace information■ SUPPORT - WorldWide Customer Support trace information
Usage Notes:	<p>Tracing assists in diagnosing unexpected or unidentifiable failures in processing the current Oracle Names server. Tracing writes a series of events from normal Oracle Names server processing to an operating system file for review by the administrator.</p> <p>Tracing output is at three levels OFF (none), USER (basic information), ADMIN (information required for administrator), or SUPPORT (information required for customer support).</p> <p>After the TRACE_LEVEL is set, tracing begins immediately. All operations are traced until it is reset to trace level OFF.</p> <p>Trace files can grow very large. Remember to turn trace level off after diagnosing the problem.</p>
Example:	<code>namesctl> set trace_level admin</code> Trace level is now 6.

SHOW CACHE_CHECKPOINT INTERVAL

Purpose:	Shows the frequency with which the Oracle Names server's cache is written to the checkpoint file.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	<p>From the operating system:</p> <pre>namesctl show cache_checkpoint_interval</pre> <p>From the NAMESCTL program:</p> <pre>namesctl> show cache_checkpoint_interval</pre>
Arguments:	None
Usage Notes:	The <code>CACHE_CHECKPOINT_INTERVAL</code> is initially set with the value in <code>NAMES.CACHE_CHECKPOINT_INTERVAL</code> in the <code>NAMES.ORA</code> file. By default, the value is 0, which disables <code>cache_checkpoint</code> . Data written to the cache checkpoint file includes net service names and addresses, and Oracle Names server addresses which were learned by the Oracle Names server as a result of forwarding a query to a foreign region on behalf of the client.
Example:	<pre>namesctl> show cache_checkpoint_interval</pre> <p>Cache checkpoint interval is currently 8 minutes 20 seconds</p>

SHOW FORWARDING_AVAILABLE

Purpose: Shows whether the Oracle Names server is forwarding requests for foreign names or redirecting them.

Prerequisites: None

Password required if one has been set: No

Syntax: From the operating system:
`namesctl show forwarding_available`
From the NAMESCTL program:
`namesctl> show forwarding_available`

Arguments: Zero or more Oracle Names servers separated by a space. If no names are given, then the setting is displayed for the current server.

Usage Notes: By default, all Oracle Names servers forward requests for foreign names. If forwarding is disabled, then requests for foreign names will be redirected to an Oracle Names server in the region which is authoritative to the requested name.

Disabling forwarding can reduce the load on a particular server, and will also make it impossible for direct clients of that server to resolve foreign names. Clients cannot be redirected, only other Oracle Names servers. See also SET FORWARDING_AVAILABLE.

Example: `namesctl> show forwarding_available`
Request forwarding is currently enabled

SHOW DEFAULT_DOMAIN

Purpose:	Displays the default domain for the NAMESCTL client.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	<p>From the operating system:</p> <pre>namesctl show default_domain</pre> <p>From the NAMESCTL program:</p> <pre>namesctl> show default_domain</pre>
Arguments:	None
Usage Notes:	<p>The existence of the DEFAULT_DOMAIN parameter allows names to be unqualified for names in that domain. For example, with DEFAULT_DOMAIN set to ACME.COM, the global name WIDE.ACME.COM could be queried using:</p> <pre>namesctl> query wide Total response time: 0.20 seconds Response status: normal, successful completion Authoritative answer: yes Number of answers: 0 TTL: 1 day</pre> <p>The initial value of DEFAULT_DOMAIN is set when the NAMESCTL program is started from the NAMES.DEFAULT_DOMAIN parameter in the SQLNET.ORA file.</p> <p>SHOW DEFAULT_DOMAIN is used when the user is unsure of the current default domain, or wants to know the default for the current configuration.</p>
Example:	<pre>namesctl> show default_domain Current default domain is "com"</pre>

SHOW LOG_FILE_NAME

Purpose:	Shows the name of the file where the Oracle Names server writes the logging information.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	From the operating system: <code>namesctl show log_file_name</code> From the NAMESCTL program: <code>namesctl> show log_file_name</code>
Arguments:	None
Usage Notes:	The LOG_FILE_NAME is initially set with the value in NAMES.LOG_FILE_NAME in the NAMES.ORA file. The default value is NAMES.LOG and is located in the \$ORACLE_HOME/network/trace on UNIX and ORACLE_HOME\network\trace on Windows NT. This file must be writable to the Oracle Names server.
Example:	<code>namesctl> show log_file_name</code> Log file name is currently <code>/private/ora23/network/names.log</code>

SHOW LOG_STATS_INTERVAL

Purpose:	Displays the frequency with which the statistics are logged to the log file.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	<p>From the operating system:</p> <pre>namesctl show [onames_server] ... log_stats_interval</pre> <p>From NAMESCTL program:</p> <pre>namesctl> show [onames_server] ... log_stats_interval</pre>
Arguments:	Zero or more Oracle Names servers separated by a space. If no names are given, then the setting is displayed for the current server.
Usage Notes:	The LOG_STATS_INTERVAL is initially set with the value in NAMES.LOG_STATS_INTERVAL in the NAMES.ORA file. By default, the value is 0, or no logging.
Example:	<pre>namesctl> show log_stats_interval Statistic counter logging is currently disabled</pre>

SHOW NAMESCTL_TRACE_LEVEL

Purpose:	Displays control of the level at which the NAMESCTL program is being traced.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	From the operating system: <code>namesctl show namesctl_trace_level</code> From the NAMESCTL program: <code>namesctl> show namesctl_trace_level</code>
Arguments:	None
Usage Notes:	Tracing assists in diagnosing unexpected or unidentifiable failures in processing the NAMESCTL program. Tracing writes a series of events from normal NAMESCTL processing to an operating system file for review by the administrator. Tracing output is at three levels OFF (none), USER (basic information), or ADMIN (maximum amount of information). SHOW NAMESCTL_TRACE_LEVEL is the only guaranteed source of what the current tracing level is.
Example:	<code>namesctl> show namesctl_trace_level</code> Controller's trace level is currently 0

SHOW REQUESTS_ENABLED

Purpose:	Shows whether or not the Oracle Names server is responding to requests.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	<p>From the operating system:</p> <pre>namesctl show [onames_server] ... requests_enabled</pre> <p>From the NAMESCTL program:</p> <pre>namesctl> show [onames_server] ... requests_enabled</pre>
Arguments:	Zero or more Oracle Names servers separated by a space. If no names are given, then the setting is displayed for the current server.
Usage Notes:	If REQUESTS_ENABLED is off, all requests to the Oracle Names server will be refused. This parameter is intended for diagnostic purposes only.
Example:	<pre>namesctl> show requests_enabled General request processing is currently enabled</pre>

SHOW RESETS_STATS_INTERVAL

Purpose:	Shows the interval set on how often the statistics are dumped to the log file.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	From the operating system: <pre>namesctl show reset_stats_interval</pre> From the NAMESCTL program: <pre>namesctl> show reset_stats_interval</pre>
Usage Notes:	If RESET_STATS_INTERVAL is initially set with the value in NAMES.RESET_STATS_INTERVAL. By default the value is set to 0, or no reset. This results in the Oracle Names server accumulating statistics the entire time it runs. For example, if statistics are reset every day, then the statistics will represent totals for the day rather than the entire time the server has been running.
Example:	<pre>namesctl> show reset_stats_interval</pre> Statistic counter reset interval is currently 5 minutes

SHOW SERVER

Purpose:	Displays the current Oracle Names server name.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	From the operating system: <pre>namesctl show server</pre> From the NAMESCTL program: <pre>namesctl> show server</pre>
Arguments:	None
Usage Notes:	SHOW SERVER displays the current Oracle Names server that commands will operate on.
Example:	<pre>namesctl> show server</pre> currently managing name server "NameServer.us.oracle.com Version banner is "Oracle Names for SunOS: Version 8.1.5.0.0"

SHOW STATUS

Purpose:	Displays the general status information about the Oracle Names server.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	<p>From the operating system:</p> <pre>namesctl show [onames_server] ... status</pre> <p>From the NAMESCTL program:</p> <pre>namesctl> show [onames_server] ... status</pre>
Arguments:	Zero or more Oracle Names servers separated by a space. If no names are given, then the setting is displayed for the current server.
Usage Notes:	Shows the current state of an Oracle Names server. Identical to the command STATUS.
Example:	<pre>namesctl> show status Version Banner is "Oracle Names for SunOS: Version 8.1.5.0.0" Server has been running for:1 day 2 hours 3 minutes 35.16 seconds....</pre>

SHOW SYSTEM_QUERIES

Purpose:	Displays the next occurrence of all system queries.
Prerequisites:	This is only relevant for distributed configurations. There are no system queries with only one administrative region.
Password required if one has been set:	No
Syntax:	From the operating system: <code>namesctl show system_queries</code> From the NAMESCTL program: <code>namesctl> show system_queries</code>
Arguments:	None
Usage Notes:	System queries are performed at intervals to keep information among Oracle Names servers current. There is no specific action that can change the activities listed as system queries. Being able to show them gives the administrator an understanding of when a system change will occur, and may assist in a decision to RESTART, thus forcing system data to be reloaded sooner.
Example:	<pre>namesctl> show system_queries System query index number:1 Query ID:49824 Query next issued in:2 hours 55 min 3.84 seconds Query state:2 Name:" " Desired data type:ns.smd</pre>

SHOW TRACE_FILE_NAME

Purpose:	Displays the trace file name and path for the current Oracle Names server.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	From the operating system: <code>namesctl show trace_file_name</code> From the NAMESCTL program: <code>namesctl> show trace_file_name</code>
Arguments:	None
Usage Notes:	The <code>TRACE_FILE_NAME</code> is initially set with the value in the <code>NAMES.TRACE_FILE_NAME</code> in the <code>NAMES.ORA</code> file. The default value is <code>NAMES.TRC</code> and is located in the <code>\$ORACLE_HOME/network/trace</code> on UNIX and <code>ORACLE_HOME\network\trace</code> on Windows NT. This file must be a valid file name, and the file must be writable to the Oracle Names server. This file is only used if tracing is enabled using the <code>NAMES.TRACE_LEVEL</code> .
Example:	<code>namesctl> show trace_file_name</code> Trace file name is currently <code>/private/ora23/network/names.trc</code>

SHOW TRACE_LEVEL

Purpose:	Displays the trace level for tracing the current Oracle Names server.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	From the operating system: <code>namesctl show trace_level</code> From the NAMESCTL program: <code>namesctl> show trace_level</code>
Arguments:	None
Usage Notes:	Tracing assists in diagnosing unexpected or unidentifiable failures in processing the current Oracle Names server. Tracing writes a series of events from normal Oracle Names server processing to an operating system file for review by the administrator. Tracing output is at three levels OFF (none), USER (basic information), or ADMIN (maximum amount of information). SHOW TRACE_LEVEL is the only guaranteed source of what the current tracing level is. Even if the TRACE_LEVEL is configured in the Oracle Names server configuration file, a previous call from the NAMESCTL program may have overridden it.
Example:	<code>namesctl> show trace_level</code> Trace level is currently 0

SHOW VERSION

Purpose:	Displays the current version and name of the Oracle Names server.
Prerequisites:	None
Password required if one has been set:	No
Syntax:	<p>From the operating system:</p> <pre>namesctl show [onames_server] ... version</pre> <p>From the NAMESCTL program:</p> <pre>namesctl> show [onames_server] ... version</pre>
Arguments:	Zero or more Oracle Names servers separated by a space. If no names are given, then the setting is displayed for the current server.
Usage Notes:	This banner identifies the server by name and version. This can be useful when clearing up minor difficulties. This command is enabled every time you connect NAMESCTL to a server.
Example:	<pre>namesctl> show version Currently managing Oracle Names server "NameServer.com" Version banner is "Oracle Names for SunOS: Version 8.1.5.0.0"</pre>

SHUTDOWN

Purpose:	Stops one or more Oracle Names servers.
Prerequisites:	Oracle Names server must be started.
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>namesctl shutdown [onames_server] ...</code> From the NAMESCTL program: <code>namesctl> shutdown [onames_server] ...</code>
Arguments:	Zero or more Oracle Names server names separated by a space. When no arguments are supplied, only the current Oracle Names server is shut down.
Usage Notes:	SHUTDOWN stops the current Oracle Names server and unloads the program from memory. an Oracle Names server should only be shut down for operational reasons like upgrades or machine maintenance. The preferred way to stop and start an Oracle Names server is using the RESTART command because you can perform it from anywhere in the network. If SHUTDOWN and START are processed individually, they must occur on the Oracle Names server machine. SHUTDOWN is identical to STOP.
Example:	<code>namesctl> shutdown</code> Confirm [yes or no] yes Server shut down.

START

Purpose:	Loads the Oracle Names service program and starts loading system and local administrative region data.
Prerequisites:	Oracle Names server must be stopped.
Password required if one has been set:	No
Syntax:	From the operating system: <code>namesctl start</code> From the NAMESCTL program: <code>namesctl> start</code>
Arguments:	None
Usage Notes:	<p>START is the command used to initially load an Oracle Names server into memory. At startup, the Oracle Names server reads its configuration files to set up its operating parameters, then loads all data for the administrative region.</p> <p>Security on Oracle Names server startup is supplied through the operating system Oracle Names is installed on. Because an Oracle Names server must be started from a local session, network security is not an issue.</p> <p>START is identical to STARTUP.</p>

START

```

Example:          namesctl> start
                  Starting "/private/dsteiner/sales/bin/names"...server
                  successfully started

                  Currently managing name server "namesrv1.us.oracle.com"
                  Version banner is "Oracle Names for Solaris: Version
                  8.1.5.0.0"

                  Server name:
                  namesrv1.us.oracle.com
                  Server has been running for:          0.16 seconds
                  Request processing enabled:           yes
                  Request forwarding enabled:           yes
                  Requests received:                    0
                  Requests forwarded:                   0
                  Foreign data items cached:            0
                  Region data next checked for reload in: not set
                  Region data reload check failures:    0
                  Cache next checkpointed in:           not set
                  Cache checkpoint interval:            not set
                  Cache checkpoint file name:
                  /private/dsteiner/sales/network/names/ckpcch.ora
                  Statistic counters next reset in:    not set
                  Statistic counter reset interval:    not set
                  Statistic counters next logged in:   not set
                  Statistic counter logging interval:  not set
                  Trace level:                          0
                  Trace file name:
                  /private/dsteiner/sales/network/trace/names_10841.trc
                  Log file name:
                  /private/dsteiner/sales/network/log/names.log
                  System parameter file name:
                  /private/dsteiner/sales/network/admin/names.ora
                  Command-line parameter file name:    ""
                  Administrative region name:          ""
                  Administrative region description:    ""
                  ApplTable Index:                     0
                  Contact                               ""
                  Operational Status                    0
                  Save Config on Stop                   yes

```

START_CLIENT_CACHE

Purpose:	Starts the client cache daemon process.
Prerequisites:	The client cache daemon process must be stopped. An Oracle Names server List must exist before you run the client cache daemon process.
Password required if one has been set:	No
Syntax:	From the operating system: <code>namesctl start_client_cache</code> From the NAMESCTL program: <code>namesctl> start_client_cache</code>
Arguments:	None
Usage Notes:	Once started, the client cache daemon process will store all information received from an Oracle Names server, making lookup faster.
Example:	<code>namesctl> start_client_cache</code>

STARTUP

Purpose:	This command is identical to START
Prerequisites:	Oracle Names server must be stopped.
Password required if one has been set:	No
Syntax:	From the operating system: <code>namesctl startup names.parameter = value</code> From the NAMESCTL program: <code>namesctl> start names.parameter = value</code>
Arguments:	None
Usage Notes:	STARTUP is identical to START .
Example:	See example for START .

STATUS

Purpose:	Displays statistics for one or more Oracle Names servers as well as many of its internal settings.
Prerequisites:	Oracle Names server must be started.
Password required if one has been set:	No
Syntax:	From the operating system: <code>namesctl status [onames_server] ...</code> From NAMESCTL program: <code>namesctl> status [onames_server] ...</code>
Arguments:	Zero or more Oracle Names server names separated by a space. When no arguments are supplied, status is given only for the current Oracle Names server.
Usage Notes:	STATUS shows the activity of the Oracle Names server over time and its state at a point in time.
Example:	<pre>namesctl> status Version banner is "Oracle Names for SunOS: 8.1.5.0.0" Server name:NSEVER.com Server has been running for:1 day 20 hours</pre>

STOP

Purpose:	Stops one or more Oracle Names servers
Prerequisites:	Oracle Names server must be started.
Password required if one has been set:	Yes If a password is set, the SET PASSWORD command must be issued prior to this command.
Syntax:	From the operating system: <code>namesctl stop [onames_server] ...</code> From the NAMESCTL program: <code>namesctl> stop [onames_server] ...</code>
Arguments:	Zero or more Oracle Names server names separated by a space. When no arguments are supplied, only the current Oracle Names server is stopped.
Usage Notes:	STOP stops the current Oracle Names server and unloads the program from memory. an Oracle Names server should only be shut down for operational reasons like upgrades or machine maintenance. The preferred way to stop and start an Oracle Names server is using the RESTART command because you can issue it from anywhere in the network. If STOP and START are processed individually, they must occur on the Oracle Names server machine. STOP is identical to SHUTDOWN.
Example:	<code>namesctl> stop</code> <code>Confirm [yes or no]: yes</code> <code>Server shut down</code>

TIMED_QUERY

Purpose:	Show all data in the Oracle Names server cache.
Password required if one has been set:	No
Syntax:	From the operating system: <code>namesctl timed_query</code> From the NAMESCTL program: <code>namesctl> timed_query [time]</code>
Usage Notes:	TIMED_QUERY returns all data since [<i>time</i>]. The <i>time</i> argument returns all objects registered after a given time. To use the <i>time</i> argument, the first TIMED_QUERY dumps out all information available since startup. At the end of the first dump is a "last timestamp" number which gives a bookmark as to where the last dump of information ended. To see all logged data since that point, provide the "last timestamp" number in the <i>time</i> argument.

UNREGISTER

Purpose:	To remove a network object from an Oracle Names server
Prerequisites:	None
Password required if one has been set:	No
Syntax:	<p>From the operating system</p> <pre>namesctl unregister <i>object_name</i> -d description=(address_list= (address= (<i>protocol_address_</i> <i>information</i>)))) [-h <i>hostname</i>] [-l <i>listener_name</i>]</pre> <p>From the NAMESCTL program:</p> <pre>namesctl> unregister <i>object_name</i> (-d <i>address_data</i>) [-h <i>hostname</i>] [-l <i>listener_name</i>]</pre>
Arguments:	Mandatory object name and the address, listener, or hostname that it was registered with.
Usage Notes:	Provides a manual mechanism for unregistering a service. The definition for that object is removed from the Oracle Names servers in the region. If the object was registered with an address, listener name, or a hostname, the address, listener name, or hostname must be provided on the command line in order to unregister the object.
Example:	<pre>namesctl> unregister parts -t oracle_database -d (description= (address= (protocol=tcp)(host=nineva)(port=1575)) (connect_ data=(service_name=db3)))</pre>

VERSION

Purpose: Displays the current version and name of the Oracle Names server

Prerequisites: None

Password required if one has been set: No

Syntax: From the operating system:
`namesctl version`
From the NAMESCTL program:
`namesctl> version`

Arguments: Zero or more Oracle Names servers separated by a space. If no names are given, then the setting is displayed for the current server

Usage Notes: This banner identifies the server by name and version. This can be useful when clearing up minor difficulties. This command is enabled every time you connect NAMESCTL to a server.

Example:
`namesctl> version`
Currently managing Oracle Names server "NameServer.com"
Version banner is "Oracle Names for SunOS: Version 8.1.5.0.0"

Oracle Connection Manager Control Utility (CMCTL)

The Oracle Connection Manager Control Utility (CMCTL) is a tool that you run from the operating system prompt to start and control Oracle Connection Manager. The general form of the Oracle Connection Manager Control Utility is:

```
CMCTL command [process_type]
```

where the *process_type* is the type of process that the command is being executed on. The choices are:

- `cman` (both **CMGW** gateway and **CMADMIN** administrative processes)
- `adm` (only the CMADMIN process)
- `cm` (only the CMGW process).

For example, to start both the administration and main processes, you would execute the following:

From the operating system:

```
cmctl start cman
```

From the CMCTL program:

```
cmctl> start cman
```

CMCTL Commands

The following commands are available through the Oracle Connection Manager Control Utility (CMCTL):

EXIT

Purpose: To exit out of the CMCTL utility program.

Prerequisites: None

Example: `cmctl> exit`

START

Purpose: To start Oracle Connection Manager.

Prerequisites: Oracle Connection Manager must not be running.

Arguments: `cm` - start both the CMGW and CMADMIN processes (default)

`cm` - start only the CMGW process

`adm` - start only the CMADMIN process

Example: `cmctl> start cm`

STATS

Purpose:	To display basic statistical information including total relays, active relays, most relays out of total, total refused.
Prerequisites:	None
Arguments:	<code>cman</code> - display statistics for both the CMGW and CMADMIN processes (default) <code>cm</code> - display statistics for the CMGW process <code>adm</code> - display statistics for the CMADMIN process
Example:	<pre>cmctl> stats (statistics=(total_relays=8)(active_delays=4)(most_ relays=6)(out_of_relay=5)(total_refused=5)</pre> <ul style="list-style-type: none">■ TOTAL_RELAYS: the total number of connections the Oracle Connection Manager has established since it started■ ACTIVE_DELAYS: the number of currently active connections■ MOST_RELAYS: the maximum number of concurrent connections the Oracle Connection Manager has ever held since it started■ OUT_OF_RELAY: the total number of connect request refusals due to out-of-relay since the Oracle Connection Manager started■ TOTAL_REFUSED: the total number of connect request refusals since the Oracle Connection Manager started

STATUS

Purpose:	To display basic information: version, start time, uptime.
Prerequisites:	None
Arguments:	<code>cman</code> - display status information for both the CMGW and CMADMIN processes (default) <code>cm</code> - display status information the CMGW process <code>adm</code> - display status information for the CMADMIN process
Example:	<pre>cmctl> status CMAN status: status=(version=8.1.5.0.0)(started=01-DEC-98 11:41:33)(state-running) ADMIN status: status=(version=8.1.5.0.0)(started=01-DEC-98 11:41:29)(state-running)</pre>

STOP

Purpose:	To stop Oracle Connection Manager processes in interactive mode.
Prerequisites:	Oracle Connection Manager must be running.
Arguments:	<code>cman</code> - stop both the CMGW and CMADMIN processes (default) <code>cm</code> - stop only the CMGW process <code>adm</code> - stop only the CMADMIN process
Usage Notes:	If you issue a stop command while connections remain active, you will be prompted to confirm the stop.
Example:	<pre>cmctl> stop cm</pre>

STOP NOW

Purpose:	To stop all Oracle Connection Manager processes. If processes are running, no prompt will confirm termination.
Prerequisites:	Oracle Connection Manager must be running.
Arguments:	None
Example:	<code>cmctl> stop now</code>

Sample Configuration Files

This appendix provides sample configuration files used in Net8. These files include:

- [Profile \(SQLNET.ORA\)](#)
- [Local Naming Configuration File \(TNSNAMES.ORA\)](#)
- [Listener Configuration File \(LISTENER.ORA\)](#)
- [Oracle Names Server Configuration File \(NAMES.ORA\)](#)
- [Oracle Connection Manager Configuration File \(CMAN.ORA\)](#)

Profile (SQLNET.ORA)

A profile (SQLNET.ORA) contains the parameters that specify preferences for how a client or server uses Net8 features. SQLNET.ORA is located in `$ORACLE_HOME/network/admin` on UNIX and `ORACLE_HOME\network\admin` on Windows platforms. For more information about each individual parameter, see "[Profile Parameters \(SQLNET.ORA\)](#)" on page C-7.

```
names.default_domain = com
names.initial_retry_timeout = 30
names.max_open_connections = 3
names.message_pool_start_size = 10
names.preferred_servers = (address_list =
  (address=(protocol=ipc)(key=n23))
  (address=(protocol=tcp)(host=nineva)(port=1383))
  (address=(protocol=tcp)(host=cicada)(port=1575))
)
names.request_retries = 2
names.directory_path = (tnsnames, onames, hostname)
namesctl.trace_directory = /oracle/network/trace
namesctl.trace_file = namesctl.trc
namesctl.trace_level = admin
namesctl.trace_unique = true
namesctl.no_initial_server = false
namesctl.internal_use = true
namesctl.noconfirm = true
namesctl.server_password = mangler
namesctl.internal_encrypt_password = false
names.dce.prefix = /./subsys/oracle/names
names.nds.name_context = personnel.acme
names.nis.meta_map=sqlnet.maps
sqlnet.authentication_services=(none)
sqlnet.authentication_services=(beq, oss)
sqlnet.kerberos5_cc_name=/tmp/mycc
sqlnet.kerberos5_clockskew=600
sqlnet.kerberos5_conf=/tmp/mykrb.conf
sqlnet.kerberos5_realms=/tmp/mykrb.realms
sqlnet.kerberos5_keytab=/tmp/myv5srvtab
sqlnet.authentication_kerberos5_service=acme
sqlnet.authentication_gssapi_service=acme/asriniva.us.oracle.com@us.oracle.com
sqlnet.identix_fingerprint_method=oracle
sqlnet.identix_fingerprint_database=ofm
sqlnet.identix_fingerprint_database_user=<username>
sqlnet.identix_fingerprint_database_password=<password>
sqlnet.authentication_gssapi_service=acme/scott.us.oracle.com@us.oracle.com
```

```
oss.source.my_wallet
=(source
  =(method=file)
    (method_data=/dve/asriniva/oss/wallet)
  )
oss.source.encrypted_private_key
=(source
  =(method=oracle)
    (method_data=
      (username=andre_security_service)
      (password=andre_security_service)
      (sqlnet_address=andreoss)
    )
  )
oss.source.certificates
=(source
  =(method=oracle)
    (method_data=
      (username=scott_security_service)
      (password=ascott_security_service)
      (sqlnet_address=andreoss)
    )
  )
oss.source.attributes
=(source
  =(method=oracle)
    (method_data=
      (username=scott_oracle_security_service)
      (password=scott_oracle_security_service)
      (sqlnet_address=andreoss)
    )
  )
sqlnet.crypto_checksum_client = required
sqlnet.encryption_client = required
sqlnet.crypto_checksum_types_client = required
sqlnet.crypto_checksum_types_server = required
sqlnet.encryption_types_client = required
sqlnet.encryption_types_server = required
sqlnet.crypto_seed = "4fhfguweotcadsfdsaf jkdsfqp5f201p45mxskdlfdasf"
sqlnet.crypto_checksum_server = required
sqlnet.encryption_server = required
trace_level_client = admin
trace_directory_client = /oracle/network/trace
trace_file_client = /oracle/network/trace/cli.trc
trace_unique_client = on
```

```
log_directory_client = /oracle/network/log
log_file_client = /oracle/network/log/sqlnet.log
log_directory_server = /oracle/network/trace
trace_directory_server = /oracle/network/trace
trace_file_server = /orace/network/trace/svr_<pid>.trc
trace_level_server = admin
use_dedicated_server = on
use_cman = true
tnsping.trace_directory = /oracle/network/trace
tnsping.trace_level = admin
sqlnet.expire_time = 10
sqlnet.client_registration = <unique_id>
bequeath_detach = yes
automatic_ipc = off
disable_oob = on
```


Local Naming Configuration File (TNSNAMES.ORA)

The local naming configuration file (TNSNAMES.ORA) contains service aliases. The service aliases of databases are mapped to connect descriptors that describe their location on the network. The TNSNAMES.ORA is used by clients and distributed database servers to identify destinations and servers. TNSNAMES.ORA is located in `$ORACLE_HOME/network/admin` on UNIX and `ORACLE_HOME\network\admin` on Windows platforms. For more information on each individual parameter, see "[Local Naming Parameters \(TNSNAMES.ORA\)](#)" on page C-31.

```
tcpnew1 = (description=
           (source_route=yes)
           (address=(protocol=tcp)(port=1610)(host=spcstn))
           (address=(protocol=tcp)(port=1580)(host=spcstn)))
           (connect_data=(service=sales1))
        )

spx2tcp = (description=
           (source_route=yes)
           (address=(protocol=spx)(service=orasrvcl))
           (address=(protocol=tcp)(port=1580)(host=spcstn)))
           (connect_data=(service_name=sales2))
        )
```

For more complex samples, see "[Local Naming Parameters \(TNSNAMES.ORA\)](#)" on page C-31.

Listener Configuration File (LISTENER.ORA)

The listener configuration file (LISTENER.ORA) contains the parameters that specify preferences for how a network listener behaves. LISTENER.ORA is located in `$ORACLE_HOME/network/admin` on UNIX and `ORACLE_HOME\network\admin` on Windows NT. For more information on each individual parameter, see "[Listener Parameters \(LISTENER.ORA\)](#)" on page C-50.

```
listener=
  (address= # default tcp listening address
   (protocol=tcp)
   (port=1521)
   (host=mudshark)
  )
  (address= # non-default ipc listening address
   (protocol=ipc)
   (key=salesdb)
  )
)
sid_list_listener=(sid_list=
  (sid_desc=
    (sid_name=sales)
    (global_dbname=salesdb.mycompany)
    (oracle_home=/privatel/app/oracle/product/8.0.3)
    (prespawn_max=20)
    (prespawn_list=
      (prespawn_desc=(protocol=tcp)(pool_size=2)(timeout=5))
      (prespawn_desc=(protocol=ipc)(pool_size=3)(timeout=2))
    )
  )
)
)
trace_level_listener=admin
trace_directory_listener=/privatel/app/oracle/product/8.0.2/network/trace
trace_file_listener=listener
logging_listener=on
log_directory_listener=/privatel/app/oracle/product/8.0.2/network/log
log_file_listener=listener
save_config_on_stop_listener=true
startup_wait_time_listener=0
```

Note: The `SID_LIST_listener_name` section is only required when making connections to a version 8 or version 7 database.

Oracle Names Server Configuration File (NAMES.ORA)

The Oracle Names server configuration file (NAMES.ORA) contains the parameters that specify the location, domain information, and optional configuration parameters for each Oracle Names server. NAMES.ORA is located in \$ORACLE_HOME/network/admin on UNIX and *ORACLE_HOME*\network\admin on Windows NT. For more information on each individual parameter, see "[Oracle Names Parameters \(NAMES.ORA\)](#)" on page C-65.

```
names.addresses = (address=(protocol=tcp)(host=oranamesvr0)(port=1575))
names.server_name = onsl.com
names.domains = (domain_list=
                 (domain=
                  (name=com)
                  (min_ttl=86400)
                 )
                 (domain=
                  (name=org)
                  (min_ttl=86400)
                 )
                )
names.admin_region= (region=
                    (name= local_region.com)
                    (type= rosdb)
                    (userid = names)
                    (password = names)
                    (description =
                     (address = (protocol = tcp)(host = nineva)(port = 1387))
                     (connect_data = (service = em))
                    )
                    (docname = sbox)
                    (version = 34619392) # 2.1.4
                    (refresh = 14400)
                    (retry = 600)
                    (expire = 259200)
                   )
names.authority_required = false
names.auto_refresh_expire = 259200
names.auto_refresh_retry = 180
names.cache_checkpoint_file = cache.ckp
names.cache_checkpoint_interval = 7200
names.config_checkpoint_file = config.ckp
names.default_forwarders=
  (forwarder_list=
   (forwarder=
```

```
(name= rootserv1.com)
(address=(protocol=tcp)(port=42100)(host=roothost)))
names.default_forwarders_only = true
names.domain_hints =
(hint_desc=
(hint_list=
(hint=(name=rootserv1.com)
(address=(protocol=tcp)(host=nineva)(port=42100))))))
names.forwarding_available = true
names.forwarding_desired = true
names.log_directory = /oracle/network/log
names.log_file = names.log
names.log_stats_interval = 3600
names.log_unique = false
names.max_open_connections = 10
names.max_reforwards = 2
names.message_pool_start_size = 24
names.no_modify_requests = false
names.no_region_database = false
names.password = 625926683431aa55
names.reset_stats_interval = 3600
names.region_checkpoint_file = reg.ckp
names.trace_directory = /oracle/network/trace
names.trace_file = names.trc
names.trace_level = admin
names.trace_unique = true
```

Oracle Connection Manager Configuration File (CMAN.ORA)

The Connection Manager configuration file (CMAN.ORA) contains the parameters that specify preferences for using Oracle Connection Manager. CMAN.ORA is located at \$ORACLE_HOME/network/admin on UNIX and *ORACLE_HOME\network\admin* on Windows NT. For more information on each individual parameter, see "[Oracle Connection Manager Parameters \(CMAN.ORA\)](#)" on page C-76

```
cmn = (address = (protocol=tcp)(host=anyhost)(port=1630))
cmn_admin = (address=(protocol=tcp)(host=anyhost)(port=1830))

cmn_profile = (parameter_list=
    (maximum_relays=512)
    (log_level=1)
    (tracing=yes)
    (trace_directory=/oracle/network/trace)
    (relay_statistics=yes)
    (show_tns_info=yes)
    (use_async_call=yes)
    (authentication_level=0)
)
# the following specifies a rule for single access control #
cmn_rules = (rule_list=
    (rule=(src=spcstn)(dst=x)(srv=x)(act=accept))
)

```

Configuration Parameters

A complete listing of all Net8 configuration parameters is provided for your reference. The following sections appear in this appendix:

- [Syntax Rules for Configuration Files](#)
- [ADDRESSes and ADDRESS_LISTs](#)
- [Profile Parameters \(SQLNET.ORA\)](#)
- [Local Naming Parameters \(TNSNAMES.ORA\)](#)
- [Listener Parameters \(LISTENER.ORA\)](#)
- [Oracle Names Parameters \(NAMES.ORA\)](#)
- [Oracle Connection Manager Parameters \(CMAN.ORA\)](#)
- [Protocol-Specific Parameters \(PROTOCOL.ORA\)](#)

Syntax Rules for Configuration Files

The configuration files in a Net8 network consist of parameters which include keyword-value pairs. Keyword-value pairs are surrounded by parentheses:

```
parameter=(keyword=value)
```

Some keywords have other keyword-value pairs as their values:

```
(keyword=  
  (keyword=value)  
  (keyword=value)  
)
```

For example, the address portion of a local naming configuration file (TNSNAMES.ORA) might include the following lines:

```
(address=  
  (protocol=tcp)  
  (host=max)  
  (port=1521)  
)
```

Set up configuration files so that indentation reflects what keyword is the parent or owner of other keyword-value pairs. This format is not required, but it does make the files much easier to read and understand.

Even if you do not choose to indent your files in this way, you must indent a wrapped line by at least one space, or it will be misread as a new parameter. The following layout is acceptable:

```
(address=(protocol=tcp)  
  (host=max) (port=1521))
```

The following layout is *not* acceptable:

```
(address=(protocol=tcp)  
(host=max) (port=1521))
```


Further Syntax Rules for Configuration Files

The following rules apply to the syntax of configuration files:

- Any keyword in a configuration file that begins a parameter that includes one or more keyword-value pairs must be in the far left column of a line. If it is indented by one or more spaces, it is interpreted as a continuation of the previous line.
- All characters must belong to the *network character set* (see the next section).
- Keywords are not case sensitive. Values may be case sensitive, depending on the operating system and protocol.
- Spaces around the "=" sign are optional in keyword-value pairs.
- There is a hierarchy of keywords in that some keywords are always followed by others. At any level of the hierarchy, keywords can be listed in any order. For example, the following entries are equally valid:

```
(address =  
  (protocol=tcp)  
  (host=martha)  
  (port=1521)  
)
```

```
(address =  
  (protocol=tcp)  
  (port=1521)  
  (host=martha)  
)
```

- Keywords cannot contain spaces. Values must not contain spaces unless enclosed within double quotes ("").
- The maximum length of a connect descriptor is 4 kilobytes.
- Comments can be included using the pound sign # at the beginning of a line. Anything following the sign to the end of the line is considered a comment.
- If the keyword-value pair consists of a single word or a concatenation of words on either side of the equal sign, no parentheses are needed.

Network Character Set

The network character set for keyword values consists of the following characters. Connect descriptors must be made up of single-byte characters.

A-Z, a-z

0-9

() < > / \

, . : ; ' " = - _

\$ + * # & ! % ? @

Within this character set, the following symbols are reserved:

() = \ " ' #

Reserved symbols are used as delimiters, not as part of a keyword or a value unless the keyword or value is quoted. Either single or double quotes can be used to enclose a value containing reserved symbols. To include a quote within a value that is surrounded by quotes, use different quote types. The backslash (\) is used as an escape character.

A specific example of the use of reserved symbols is a numeric DECnet object within an address. An OBJECT can be a name such as ABC or #123. These would be entered in the form:

(OBJECT=ABC)

or

(OBJECT=\#123)

Because the "#" sign is a reserved symbol, the character # must be preceded by a backslash.

The following characters may be used within a connect descriptor, but not in a keyword or value:

<space> <tab> <CR> <newline>

Character Set

The listener name, net service name, and Oracle Connection Manager names are limited to the following character set:

[a...z] [A...Z] [0...9] _

The first character must be an alphabetical character. In general, up to 64 characters is acceptable. A database service name must match the global database name defined by the database administrator, which consists of a database name (originally limited to eight characters), and the database domain. Net service names and global database names are not case sensitive.

ADDRESSES and ADDRESS_LISTS

When a connection is made, the client and the receiver of the request, such as a listener, Oracle Names server, or Oracle Connection Manager, are configured with identical protocol addresses. The purpose of the address is to identify the network address of a network object. The client uses this address to send the connection request to a particular network object location, and the recipient "listens" for requests on this address, and grants a connection based on its address information matching the client information. It is important to install the same protocols for the client and the connection recipient, as well as configure the same address.

Protocol address are comprised of two elements:

ADDRESS

Purpose: Defines a protocol address. This parameter may be embedded under ADDRESS_LIST or DESCRIPTION. A DESCRIPTION is typically only used in a TNSNAMES.ORA or LISTENER.ORA file.

Additional Information: Each protocol has its own required parameters needed in the address, as explained on "[Configuring Protocol Addresses](#)" on page 6-76.

Example:

```
(address=  
  (protocol=tcp)  
  (host=sales-pc)  
  (port=1521)  
)
```

ADDRESS_LIST

Purpose: Defines lists of addresses that share a set of common characteristics.

This parameter may be embedded under DESCRIPTION or DESCRIPTION_LIST. A DESCRIPTION or DESCRIPTION_LIST is typically only used in a TNSNAMES.ORA or LISTENER.ORA file.

Note: If there is only one list in a TNSNAMES.ORA or a LISTENER.ORA file, ADDRESS_LIST is not necessary.

Example:

```
(address_list=  
  (address=  
    (protocol=tcp)  
    (host=sales-pc)  
    (port=1521)  
  )  
  (address=  
    (protocol=tcp)  
    (host=hr-pc)  
    (port=1521)  
  )  
)  
(address_list=  
  (address=  
    (protocol=tcp)  
    (host=finance-pc)  
    (port=1521)  
  )  
  (address=  
    (protocol=tcp)  
    (host=sales-pc)  
    (port=1521)  
  )  
)  
)
```

Profile Parameters (SQLNET.ORA)

The following parameters are available in a profile. Profiles are stored in a configuration file called SQLNET.ORA. SQLNET.ORA is located in `$ORACLE_HOME/network/admin` on UNIX and `ORACLE_HOME\network\admin` on Windows platforms.

BEQUEATH_DETACH

Purpose: Turn on/off signal handling in Net8 on UNIX systems.

Default: NO which leaves signal handling on.

Values:

- YES - Turns off signal handling
- NO - Leaves signal handling on

Example: `bequeath_detach=yes`

DAEMON.TRACE_DIRECTORY

Purpose: Controls the destination directory of the Oracle Enterprise Manager daemon trace file.

Default: `$ORACLE_HOME/network/trace` on UNIX platforms and `ORACLE_HOME/network/trace` on Windows platforms

Example: `daemon.trace_directory=/oracle/traces`

DAEMON.TRACE_LEVEL

Purpose: Turns tracing on/off to a certain specified level for the Oracle Enterprise Manager daemon.

Default: OFF

Values:

- OFF - No trace output
- USER - User trace information
- ADMIN - Administration trace information
- SUPPORT - WorldWide Customer Support trace information

Example: `daemon.trace_level=user`

DAEMON.TRACE_MASK

Purpose: Specifies that only the Oracle Enterprise Manager daemon trace entries are logged into the trace file.

Default: `$ORACLE_HOME/network/trace` on UNIX platforms and `ORACLE_HOME/network/trace` on Windows platforms

Example: `daemon.trace_mask=(106)`

DISABLE_OOB

Purpose: Disables out of band breaks.

Default: OFF

Usage Notes: Disable out of band breaks if the underlying transport protocol does not support the feature.

Example: `disable_oob=off`

LOG_DIRECTORY_CLIENT

Purpose: Controls the directory for where the log file is written.

Default: Current directory where executable is started from.

Example: `log_directory_client=/oracle/network/log`

LOG_DIRECTORY_SERVER

Purpose: Controls the directory for where the log file is written.

Default: Current directory where executable is started from.

Example: `log_directory_server=/oracle/network/log`

LOG_FILE_CLIENT

Purpose: Controls the log output file name for an Oracle client.
Default: SQLNET.LOG
Example: log_file_client=client

LOG_FILE_SERVER

Purpose: Controls the log output file name for an Oracle server.
Default: SQLNET.LOG
Example: log_file_server=svr.log

NAMES.DCE.PREFIX

Purpose: Specifies the DCE cell name (prefix) to use for name lookup.
Default: /./subsys/oracle/names
Example: names.dce.prefix=/./subsys/oracle/names

NAMES.DEFAULT_DOMAIN

Purpose: Indicates the domain from which the client most often requests names. When this parameter is set, the default domain name will be automatically appended to any unqualified service name. Any name which contains an unescaped dot ('.') will not have the default domain appended. Simple names may be qualified with a trailing dot (for example, 'rootserver.').
Default: NULL
Example: names.default_domain=com

NAMES.DIRECTORY_PATH

Purpose: Indicates the order of the naming methods, such as TNSNAMES or Oracle Names, that will be used for client name requests.

Default: TNSNAMES, ONAMES, HOSTNAME

- Values:**
- TNSNAMES
 - ONAMES
 - HOSTNAME
 - DCE
 - NIS
 - NOVELL

Example: `names.directory_path=(tnsnames, onames)`

NAMES.INITIAL_RETRY_TIMEOUT

Purpose: Determines how long a client will wait for a response from an Oracle Names server before reiterating the request to the next server in the preferred servers list.

Default: 15

Minimum Value: 1

Maximum Value: 600

Example: `names.initial_retry_timeout=20`

NAMES.MAX_OPEN_CONNECTIONS

Purpose:	Determines how many connections an Oracle Names client may have open at one time.
Default:	10
Minimum Value:	3
Maximum Value:	64
Example:	<code>names.max_open_connections=3</code>

NAMES.MESSAGE_POOL_START_SIZE

Purpose:	Determines the initial number of messages allocated in the client's message pool which are used for forwarded message requests.
Default:	10
Minimum Value:	3
Maximum Value:	256
Example:	<code>names.message_pool_start_size=10</code>

NAMES.NIS.META_MAP

Purpose: Specifies the file to be used to map NIS attributes to an NIS mapname.

Default: SQLNET.MAPS

Example: `names.nis.meta_map=sqlnet.maps`

NAMES.PREFERRED_SERVERS

Purpose: Indicates the name, addresses, and order of Oracle Names servers that will be used for a client's name requests.

Default: None

Example: `names.preferred_servers= (address_list=
(address=(protocol=ipc)(key=n23))
(address=(protocol=tcp)(host=nineva)(key=1575))
(address=(protocol=tcp)(host=cicada)(key=1575)))`

NAMES.REQUEST_RETRIES

Purpose: Specifies the number of times the client should try each server in the list of preferred servers before allowing the operation to fail.

Default: 1

Minimum Value: 1

Maximum Value: 5

Example: `names.request_retries=5`

NAMESCTL.INTERNAL_ENCRYPT_PASSWORD

Purpose:	If set to TRUE, NAMESCTL encrypts the password when it is sent to the Oracle Names server. If set to FALSE, NAMESCTL does not encrypt the password when it is sent to the Oracle Names server. This enables unencrypted passwords to be set in the NAMES.ORA file with the NAMES.PASSWORD parameter.
Default:	FALSE
Values:	TRUE FALSE
Example:	<code>namesctl.internal_encrypt_password=true</code>

NAMESCTL.INTERNAL_USE

Purpose:	If set to TRUE, NAMESCTL enables a set of internal undocumented commands. All internal commands are preceded by an underscore in order to distinguish them as internal.
----------	---

NAMESCTL.NO_INITIAL_SERVER

Purpose:	If set to TRUE, NAMESCTL suppresses any error messages when unable to connect to a default Oracle Names server.
Default:	FALSE
Values:	TRUE FALSE
Example:	<code>namesctl.no_initial_server=true</code>

NAMESCTL.NOCONFIRM

Purpose:	Indicates whether sensitive commands (STOP, RELOAD, RESTART) should be prompted with a confirmation when running the NAMESCTL utility.
Default:	OFF
Values:	ON OFF
Example:	<code>namesctl.noconfirm=on</code>

NAMESCTL.SERVER_PASSWORD

Purpose: Indicates the value that matches the configured password set in the NAMES.ORA file with the **NAMES.PASSWORD** parameter. This eliminates the need to enter the password with the **SET PASSWORD** command each time you use the NAMESCTL utility to perform secure options, such as STOP, RESTART and RELOAD.

Example: `namesctl.server_password=secret`

NAMESCTL.TRACE_LEVEL

Purpose: Indicates the level at which the NAMESCTL program should be traced.

Default: OFF

Values: OFF, USER, ADMIN, SUPPORT

Example: `namesctl.trace_level=admin`

NAMESCTL.TRACE_FILE

Purpose: Indicates the file in which the NAMESCTL trace output is placed.

Default: `NAMESCTL_PID.TRC`

Example: `namesctl.trace_file=nmsctl`

NAMESCTL.TRACE_DIRECTORY

Purpose: Indicates the directory where trace output from the NAMESCTL utility is placed.

Default: `$ORACLE_HOME/network/trace` on UNIX and `ORACLE_HOME/network/trace` on Windows NT

Example: `namesctl.trace_directory=/oracle/trace`

NAMESTL.TRACE_UNIQUE

Purpose:	Indicates whether a process identifier is appended to the name of each trace file generated, so that several can co-exist.
Default:	ON
Values:	ON OFF
Example:	<code>namesctl.trace_unique=on</code>

OSS.SOURCE.LOCATION

Purpose:	Defines the method for retrieving encrypted private keys.
Default:	None
Example:	<code>oss.source.location= (source= (method=oracle) (method_data= (sqlnet_address=oss)))</code>

OSS.SOURCE.MY_WALLET

Purpose:	Specifies the location of wallets. Wallets are certificates, keys and trustpoints processed by SSL.
Default:	None
Example:	<code>oss.source.my_wallet= (source= (method=file) (method_data= (directory=/home/smalladi/oss)))</code>

SQLNET.AUTHENTICATION_SERVICES

Purpose:	Enables one or more authentication services. If authentication has been installed, it is recommended that this parameter be set to either NONE or to one of the authentication methods.
Default:	None
Values:	NONE - the database will use the user name and password to log in ALL - Enables all authentication methods to be used. BEQ - always used in conjunction with one or more of the following items: NDS - use NetWare Directory Services (NDS) NTS - use Windows NT Native KERBEROS5 - use Kerberos SECURID - use SecurID CYBERSAFE - use Cybersafe IDENTIX - use Identix DCEGSSAPI - use DCE GSSAPI RADIUS - use RADIUS
Example:	<code>sqlnet.authentication_services =(beq, kerberos5, cybersafe)</code>

SQLNET.AUTHENTICATION_KERBEROS5_SERVICE

Purpose:	Defines the name of the service used to obtain a Kerberos service ticket
Default:	None
Example:	<code>sqlnet.authentication_kerberos5_service= oracle</code>

SQLNET.AUTHENTICATION_GSSAPI_SERVICE

Purpose:	Defines the CyberSAFE service principal
----------	---

SQLNET.CLIENT_REGISTRATION

Purpose:	Sets a unique identifier for this client machine. This identifier will be passed to the listener with any connection request and will be included in the Audit Trail. The identifier can be any alphanumeric string up to 128 characters long.
Default:	None
Example:	<code>sqlnet.client_registration 1432</code>

SQLNET.CRYPTO_CHECKSUM_CLIENT

Purpose:	Specifies the desired checksum behavior when this client is connecting to a server
Default:	ACCEPTED
Values:	<ul style="list-style-type: none">■ ACCEPTED■ REJECTED■ REQUESTED■ REQUIRED
Example:	<code>sqlnet.crypto_checksum_client=accepted</code>

SQLNET.CRYPTO_CHECKSUM_SERVER

Purpose:	Specifies the desired checksum behavior when a client is connecting to this server
Default:	ACCEPTED
Values:	<ul style="list-style-type: none">■ ACCEPTED■ REJECTED■ REQUESTED■ REQUIRED
Example:	<code>sqlnet.crypto_checksum_server=accepted</code>

SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT

Purpose:	Specifies a list of crypto-checksum algorithms this client is allowed to use
Default:	MD5
Values:	MD5 - RSA Data Security's MD5 algorithm
Example:	<code>sqlnet.crypto_checksum_types_client=(md5)</code>

SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER

Purpose:	Specifies a list of crypto-checksum algorithms this server is allowed to use
Default:	MD5
Values:	MD5 - RSA Data Security's MD5 algorithm
Example:	<code>sqlnet.crypto_checksum_types_server=(md5)</code>

SQLNET.CRYPTO_SEED

Purpose: Specifies the characters used when generating cryptographic keys. The more random the characters are, the stronger the keys are. This is required whenever encryption or checksumming is turned on. The string should be 10-70 random characters.

Default: qwertyuiopasdfghjkl;zxcvbnm,.s1

Example: `sqlnet.crypto_seed="qwertyuiopasdfghjkl;zxcvbnm,.s1"`

SQLNET.ENCRYPTION_CLIENT

Purpose: Specifies the desired behavior when this client is connecting to a server

Default: ACCEPTED

Values:

- ACCEPTED
- REJECTED
- REQUESTED
- REQUIRED

Example: `sqlnet.encrypted_client=accepted`

SQLNET.ENCRYPTION_SERVER

Purpose: Specifies the desired behavior when a client is connecting to this server

Default: ACCEPTED

Values:

- ACCEPTED
- REJECTED
- REQUESTED
- REQUIRED

Example: `sqlnet.encrypted_server=accepted`

SQLNET.ENCRYPTION_TYPES_CLIENT

Purpose:	Specifies a list of encryption algorithms this client is allowed to use
Default:	All algorithms are used if none are specified.
Values:	One or more of the following: RC4_40 - RSA RC4 (40 bit key size) Domestic & International RC4_56 - RSA RC4 (56 bit key size) Domestic only RC4_128 - RSA RC4 (128 bit key size) Domestic only DES - Standard DES (56 bit key size) Domestic only DES40 - (40 bit key size) Domestic & International
Example:	<code>sqlnet.encryption_types_client=(rc4_40)</code>

SQLNET.ENCRYPTION_TYPES_SERVER

Purpose:	Specifies a list of encryption algorithms this server is allowed to use when acting as a server
Default:	All algorithms are used if none are specified.
Values:	RC4_40 - RSA RC4 (40 bit key size) Domestic & International RC4_56 - RSA RC4 (56 bit key size) Domestic only RC4_128 - RSA RC4 (128 bit key size) Domestic only DES - Standard DES (56 bit key size) Domestic only DES40 - (40 bit key size) Domestic & International
Example:	<code>sqlnet.encryption_types_server=(rc4_40, des, ...)</code>

SQLNET.EXPIRE_TIME

Purpose:	Determines time interval to send a probe to verify the session is alive
Default:	None
Minimum Value:	0 minutes
Recommended Value:	10 minutes
Example:	<code>sqlnet.expire_time=10</code>

SQLNET.IDENTIX_FINGERPRINT_DATABASE

Purpose: Specifies the service name or alias for the authentication fingerprint database

Default: None

Example: `sqlnet.identix_fingerprint_database=fingrdb`

SQLNET.IDENTIX_FINGERPRINT_DATABASE_USER

Purpose: Specifies the well known user name for the fingerprint database

Default: None

Example: `sqlnet.identix_fingerprint_database_user=manager`

SQLNET.IDENTIX_FINGERPRINT_DATABASE_PASSWORD

Purpose: Specifies the well known password for the fingerprint database

Default: None

Example: `sqlnet.identix_fingerprint_database_password=password`

SQLNET.IDENTIX_FINGERPRINT_METHOD

Purpose: Specifies the method name for the fingerprint database. The method name must be ORACLE

Default: None

Example: `sqlnet.identix_fingerprint_method=oracle`

SQLNET.KERBEROS5_CC_NAME

Purpose: Specifies the complete path name to the Kerberos credentials cache file.

Default: /usr/tmp/krbcache on UNIX and c:\tmp\krbcache on Windows platforms

Example: sqlnet.kerberos5_cc_name= /usr/tmp/krbcache

SQLNET.KERBEROS5_CLOCKSKEW

Purpose: Specifies how many seconds can pass before a Kerberos credential is considered out of date.

Default: 300

Example: sqlnet.kerberos5_clockskew=1200

SQLNET.KERBEROS5_CONF

Purpose: Specifies the complete path name to the Kerberos configuration file, which contains the realm for the default KDC and maps realms to KDC hosts.

Default: /krb5/krb.conf on UNIX and c:\krb5\krb.conf on Windows platforms

Example: sqlnet.kerberos5_conf=/krb5/krb.conf

SQLNET.KERBEROS5_KEYTAB

Purpose:	Specifies the complete path name to the Kerberos principal/secret key mapping file, which is used to extract keys and decrypt incoming authentication information.
Default:	<code>/etc/v5srvtab</code> on UNIX and <code>c:\krb5\v5srvtab</code> on Windows platforms
Example:	<code>sqlnet.kerberos5_keytab=/etc/v5srvtab</code>

SQLNET.KERBEROS5_REALMS

Purpose:	Specifies the complete pathname to the Kerberos realm translation file, which provides a mapping from a host name or domain name to a realm.
Default:	<code>/krb5/krb.realms</code> on UNIX and <code>c:\krb5\krb.realms</code> on Windows platforms
Example:	<code>sqlnet.kerberos5_realms= /krb5/krb.realms</code>

SQLNET.RADIUS_ALTERNATE

Purpose:	Specifies an alternate RADIUS server used in case the primary server is unavailable. The value can be either the IP address or host name of the server.
Default:	None
Example:	<code>sqlnet.radius_alternate=radius2</code>

SQLNET.RADIUS_ALTERNATE_PORT

Purpose:	Listening port of the alternate RADIUS server.
Default:	1645
Example:	<code>sqlnet.radius_alternate_port= 1667</code>

SQLNET.RADIUS_ALTERNATE_RETRIES

Purpose: Times to resend
Default: 3
Example: `sqlnet.radius_alternate_retries=4`

SQLNET.RADIUS_AUTHENTICATION_TIMEOUT

Purpose: Time to wait for response.
Default: 5 seconds
Example: `sqlnet.radius_authentication_timeout= 10`

SQLNET.RADIUS_AUTHENTICATION

Purpose: Specifies the location of the primary RADIUS server, either by its host name or IP address.
Default: local host
Example: `sqlnet.radius_authentication= officeacct`

SQLNET.RADIUS_AUTHENTICATION_INTERFACE

Purpose: Specifies the class containing the user interface used to interact with the user.
Default: DefaultRadiusInterface
Example: `sqlnet.radius_authentication_interface=`

SQLNET.RADIUS_AUTHENTICATION_PORT

Purpose: Listening port of the primary RADIUS server.
Default: 1645
Example: `sqlnet.radius_authentication_port= 1667`

SQLNET.RADIUS_AUTHENTICATION_RETRIES

Purpose: Times to resend.
Default: 3
Example: `sqlnet.radius_authentication_retries=4`

SQLNET.RADIUS_AUTHENTICATION_TIMEOUT

Purpose: Time to wait for response.
Default: 5 seconds
Example: `sqlnet.radius_authentication_timeout= 10`

SQLNET.RADIUS_CHALLENGE_RESPONSE

Purpose: Turns challenge response ON or OFF.
Default: OFF
Values: ON | OFF
Example: `sqlnet.radius_challenge_response=ON`

SQLNET.RADIUS_SECRET

Purpose: Specifies the full path to the file containing the RADIUS shared secret.
Default: `ORACLE_HOME/network/security/radius.key` on UNIX and `ORACLE_HOME/network/security/radius.key` on Windows NT
Example: `sqlnet.radius_secret= oracle/bin/admin/radiuskey`

SQLNET.RADIUS_SEND_ACCOUNTING

Purpose:	Turns accounting ON and OFF. If enabled, packets will be sent to the active RADIUS server at listening port plus one. The default port is 1646.
Default:	OFF
Values:	ON OFF
Example:	<code>sqlnet.radius_send_accounting=ON</code>

SSL_CLIENT_AUTHENTICATION

Purpose:	Controls whether or not a client—in addition to the server—is authenticated using SSL.
Default:	TRUE
Values:	TRUE FALSE
Example:	<code>ssl_cipher_suite=(ssl_dh_dss_with_des_cdc_sha)</code>

SSL_CIPHER_SUITES

Purpose:	Controls what combination of encryption and data integrity is used by SSL.
Default:	None
Values:	For further information about Cipher Suites, see the <i>Oracle Advanced Security Administrator's Guide</i> .
Example:	<code>ssl_cipher_suite=(ssl_rsa_with_rc4_138_md5)</code>

SSL_VERSION

Purpose:	Forces the version of the SSL connection. Clients and servers must use a compatible version.
Default:	UNDETERMINED
Values:	UNDETERMINED 2.0 3.0
Example:	<code>ssl_version=2.0</code>

TNSPING.TRACE_DIRECTORY

Purpose: Controls the destination directory of the trace file.
Default: \$ORACLE_HOME/network/trace
Example: tnsping.trace_directory=/oracle/traces

TNSPING.TRACE_LEVEL

Purpose: Turns tracing on/off to a certain specified level.
Default: OFF
Values

- OFF - No trace output
- USER - User-induced error conditions trace information
- ADMIN - Administration trace information
- SUPPORT - WorldWide Customer Support trace information

Example: tnsping.trace_level=admin

TRACE_DIRECTORY_CLIENT

Purpose: Controls the destination directory of the trace file.
Default: \$ORACLE_HOME/network/trace
Example: trace_directory_client=/oracle/traces

TRACE_DIRECTORY_SERVER

Purpose: Controls the destination directory of the trace file.
Default: \$ORACLE_HOME/network/trace
Example: trace_directory_server=/oracle/traces

TRACE_FILE_CLIENT

Purpose: Controls the name of the client trace file.
Default: SQLNET.TRC

TRACE_FILE_CLIENT

Example: `trace_file_client=clientsqlnet.trc`

TRACE_FILE_SERVER

Purpose: Controls the name of the server trace file.

Default: `SVR_PID.TRC`

Example: `trace_file_server=svrsqlnet.trc`

TRACE_LEVEL_CLIENT

Purpose: Turns tracing on/off to a certain specified level.

Default: `OFF`

Values

- `OFF` - No trace output
- `USER` - User trace information
- `ADMIN` - Administration trace information
- `SUPPORT` - WorldWide Customer Support trace information

Example: `trace_level_client=user`

TRACE_LEVEL_SERVER

Purpose:	Turns tracing on/off to a certain specified level.
Default:	OFF
Values	<ul style="list-style-type: none">■ OFF - No trace output■ USER - User trace information■ ADMIN - Administration trace information■ SUPPORT - WorldWide Customer Support trace information
Example:	<code>trace_level_server=admin</code>

TRACE_UNIQUE_CLIENT

Purpose:	<p>Used to make each client trace file have a unique name to prevent each trace file from being overwritten with the next occurrence of the client. The PID is attached to the end of the file name.</p> <p>Determines whether or not a unique trace file is created for each client. The same trace file name is used for every client. If the value is OFF, when a new trace file is created for a client, it overwrites the existing file. If the value is set to ON, a thread identifier is appended to the name of each trace file generated so that several can coexist.</p> <p>This parameter creates unique trace files named <code>SQLNETTHREAD_ID.TRC</code>.</p>
Default:	ON
Example:	<code>trace_unique_client=on</code>

USE_CMAN

Purpose:	When set to TRUE, forces all sessions to go through Oracle Connection Manager to get to the server. If no Oracle Connection Manager addresses are available, connections are routed through any available listener address. Note: If you are using Oracle Connection Manager with Oracle Names, this option must be set on clients and Oracle Names servers.
Default:	FALSE
Values:	TRUE FALSE
Example:	<code>use_cman=true</code>

USE_DEDICATED_SERVER

Purpose:	Forces the listener to spawn a dedicated server process for sessions from this client.
Default:	OFF
Values:	<ul style="list-style-type: none">■ ON - spawn dedicated server processes■ OFF - hand off to existing server processes
Example:	<code>use_dedicated_server=on</code>

Local Naming Parameters (TNSNAMES.ORA)

With the local naming method, net service names are added to the TNSNAMES.ORA file. TNSNAMES.ORA is located in `$ORACLE_HOME/network/admin` on UNIX and `ORACLE_HOME\network\admin` on Windows platforms.

A net service name is an alias mapped to a database network address contained in a connect descriptor. A connect descriptor contains the location of the listener through a protocol address and the service name of the database to which to connect. Clients and servers that are clients of other servers use this net service name when making a connection with an application.

General Syntax

The basic syntax for a TNSNAMES.ORA file's is shown in [Figure C-1](#). DESCRIPTION contains the connect descriptor, ADDRESS is the listener address, and CONNECT_DATA contains the database service name.

Figure C-1 General Syntax of TNSNAMES.ORA

```
net_service_name=  
(description=  
  (address= (protocol_address_information))  
  (connect_data=  
    (service_name=service_name)  
  )  
)
```

Multiple Descriptions

A TNSNAMES.ORA file may contain one or more connect descriptors. Each connect descriptor can contain one or more listener addresses. Shown below in [Figure C-2](#) are two connect descriptors with multiple addresses. DESCRIPTION_LIST defines a list of connect descriptors.

Note: The Net8 Assistant does not support multiple connect descriptors.

Figure C-2 Multiple Descriptions in TNSNAMES.ORA

```
net_service_name=
(description_list=
  (description=
    (address= (protocol_address_information))
    (address= (protocol_address_information))
    (address= (protocol_address_information))
    (connect_data=
      (service_name=service_name)
    )
  )
  (description=
    (address= (protocol_address_information))
    (address= (protocol_address_information))
    (address= (protocol_address_information))
    (connect_data=
      (service_name=service_name)
    )
  )
)
```

Multiple Address Lists

The TNSNAMES.ORA also supports multiple list of addresses, each with its own characteristics. In [Figure C-3](#), two address lists are presented. The first address list features client load balancing and no connect-time failover, affecting only those listener ADDRESSES within the ADDRESS_LIST. The second address list features connect-time failover and no client load loading balance, affecting only those listener ADDRESSES within the ADDRESS_LIST. The client will first try either the first or second address at random, then try address three and four sequentially.

Note: The Net8 Assistant only supports the creation of one address list.

Figure C-3 Multiple Address Lists in TNSNAMES.ORA

```
net_service_name=
  (description=
    (address_list=
      (load_balance=on)
      (failover=off)
      (address= (protocol_address_information))
      (address= (protocol_address_information))
    )
    (address_list=
      (load_balance=off)
      (failover=on)
      (address= (protocol_address_information))
      (address= (protocol_address_information))
    )
    (connect_data=
      (service_name=service_name)
    )
  )
)
```

Note: Address lists do not have to be embedded in an ADDRESS_LIST if there is only one list, as was the case prior to release 8.1.

Changed Functionality of Client Load Balancing and DESCRIPTION_LISTS

Prior to release 8.1, DESCRIPTION_LISTS were used for client load balancing of multiple listeners. While client load balancing is still on by default for DESCRIPTION_LISTS, client load balancing can also be explicitly specified for an ADDRESS_LIST or associated with a set of ADDRESSES or set DESCRIPTIONS. Shown in [Figure C-4](#) is a comparison of a TNSNAMES.ORA prior to release 8.1 and for release 8.1, using client load balancing in an Oracle Parallel Sever environment:

Figure C-4 Client Load Balancing for Release 8.1 and Prior to Release 8.1**Client Load Balancing Prior to Release 8.1**

Shown below is pre-8.1 release TNSNAMES.ORA file with net service names of OP, OP1 and OP2.

OPS allows a client to randomly choose one of the two instances. OP1 and OP2 allow a client to connect to a specific Oracle Parallel Server instance.

```
op=
  (description_list =
    (description=
      (address=
        (protocol=tcp)
        (host=opsnt1)
        (port=1521))
      (connect_data=(sid=op1)))

    (description=
      (address=
        (protocol=tcp)
        (host=opsnt2)
        (port=1521))
      (connect_data=(sid=op2))))

op1=
  (description=
    (address=
      (protocol=tcp)
      (host=opsnt1)
      (port=1521))
    (connect_data=(sid=op1)))

op2=
  (description=
    (address=
      (protocol=tcp)
      (host=opsnt1)
      (port=1521))
    (connect_data=(sid=op2)))
```

Client Load Balancing in Release 8.1

Shown below is a 8.1 release TNSNAMES.ORA file with net service names of OP, OP1 and OP2.

OP allows a client to randomly to connect to database service OP and choose one of the two instances using multiple ADDRESSES rather than multiple DESCRIPTIONS. OP1 and OP2 allow a client to connect to a specific Oracle Parallel Server instance.

```
op=
  (description=
    (load_balance=on)
    (address=
      (protocol=tcp)
      (host=opsnt1)
      (port=1521))
    (address=
      (protocol=tcp)
      (host=opsnt2)
      (port=1521))
    (connect_data=
      (service_name=op)))

op1=
  (description=
    (address=
      (protocol=tcp)
      (host=opsnt1)
      (port=1521))
    (connect_data=
      (service_name=op)
      (instance_name=op1)))

op2=
  (description=
    (address=
      (protocol=tcp)
      (host=opsnt2)
      (port=1521))
    (connect_data=
      (service_name=op)
      (instance_name=op2)))
```

Examples

When a client goes through an Oracle Connection Manager for a connection to a service, the first address (or set of address) is for a client connection to an Oracle Connection Manager and the second address (or set of addresses) is for an Oracle Connection Manager connection to a service using listener.

When the Oracle Connection Manager parameter `SOURCE_ROUTE` is set to `YES`, it instructs the client to go through all the addresses listed before connecting.

Shown in [Figure C-5](#) is an example with multiple Oracle Connection Manager addresses:

Figure C-5 Multiple Oracle Connection Manager Addresses TNSNAMES.ORA

```
sample1=(description=
  (source_route=yes)
  (address=(protocol=tcp)(host=host1)(port=1630))    # hop 1
  (address_list=
    (failover=on)
    (load_balance=off)                                # hop 2
    (address=(protocol=tcp)(host=host2a)(port=1630))
    (address=(protocol=tcp)(host=host2b)(port=1630))
  )
  (address=(protocol=tcp)(host=host3)(port=1521))    # hop 3
  )
  (connect_data=(service_name=sales.acme.com))
)
```

In the example above:

1. The client is instructed to connect to an address of the first Oracle Connection Manager, as indicated by:

```
(address=(protocol=tcp)(host=host1)(port=1630))
```

2. The first Oracle Connection Manager then connects to another Oracle Connection Manager by first trying the address of one Oracle Connection Manager. If the first address fails, it tries the second address. This is indicated by:

```
(address_list=
  (failover=on)
  (load_balance=off)                                # hop 2
  (address=(protocol=tcp)(host=host2a)(port=1630))
  (address=(protocol=tcp)(host=host2b)(port=1630))
)
```

3. The Oracle Connection Manager then connects to the database service using the following listener address:

```
(address=(protocol=tcp)(host=host3)(port=1521)) # hop 3
```

Shown in [Figure C-6](#) is an example of client load balancing among two Oracle Connection Managers and two listener addresses:

Figure C-6 Client Load Balancing in TNSNAMES.ORA

```
sample2=(description=
  (load_balance=on)
  (failover=on)
  (address_list=
    (source_route=yes)
    (address=(protocol=tcp)(host=host1)(port=1630))
    (address=(protocol=tcp)(host=host2)(port=1521))
  )
  (address_list=
    (source_route=yes)
    (address=(protocol=tcp)(host=host3)(port=1630))
    (address=(protocol=tcp)(host=host4)(port=1521))
  )
  (connect_data=(service_name=sales.acme.com))
)
```

In the example above:

1. The client is instructed to pick an ADDRESS_LIST at random and to failover to the other if the chosen ADDRESS_LIST fails. This is indicated by the LOAD_BALANCE and FAILOVER parameters being set to ON.
2. When an ADDRESS_LIST is chosen, the client first connects to the Oracle Connection Manager, using the Oracle Connection Manager address indicated for the ADDRESS_LIST.
3. The Oracle Connection Manager then connects to the database service, using the listener address indicated for the ADDRESS_LIST.

DESCRIPTIONS and DESCRIPTION_LISTs

DESCRIPTION

Purpose: Beginning of a connect descriptor, containing the definition of a database listening address and the service name to which to connect. This parameter may be embedded under a DESCRIPTION_LIST.

Example:

```
net_service_name=  
(description =  
  (address=...)  
  (connect_data=(service_name=sales.com))
```

DESCRIPTION_LIST

Purpose: Defines a list of connect descriptors for a particular net service name.

Example:

```
net_service_name=  
(description_list=  
  (description=  
    (address=...)  
    (connect_data=(service_name=sales.com))  
  (description=  
    (address=...)  
    (connect_data=(service_name=sales2.com))  
  )
```

ADDRESSES and ADDRESS_LISTS

ADDRESS

Purpose: Defines a listener protocol address. This parameter may be embedded under a ADDRESS_LIST or DESCRIPTION.

Additional Information: See "[Configuring Protocol Addresses](#)" on page 6-76 for descriptions of the correct parameters to use for each protocol.

Example:

```
net_service_name=
(description=
  (address =...)
  (address =...)
  )
(connect_data=(service_name=sales.com))
```

ADDRESS_LIST

Purpose: Defines a list of addresses. If there is only list, ADDRESS_LIST is not necessary. This parameter may be embedded under a DESCRIPTION or DESCRIPTION_LIST.

Example:

```
net_service_name=
(description=
  (address_list=
    (address =...)
    (address =...)
    )
  )
  (address_list=
    (load_balance=off)
    (address =...)
    (address =...)
    )
  )
(connect_data=(service_name=sales.com))
```

Optional Parameters for Lists

FAILOVER

Purpose: When set to ON, instructs Net8 at connect time to fail over to a different listener if the first listener fails. It determines how many addresses are tried. When set to OFF, instructs Net8 to try one address.

This parameter must be embedded under a DESCRIPTION_LIST, DESCRIPTION or ADDRESS_LIST.

Note: This feature will not work if the LISTENER.ORA is configured with static service configuration parameters. See "[SID_LIST_listener_name Static Service Section](#)" on page C-57 for further information about these parameters. Static configuration is required for Oracle8i release 8.0 or Oracle7 databases and Oracle Enterprise Manager.

Default: ON for DESCRIPTION_LISTs, DESCRIPTIONs, and ADDRESS_LISTs.

Values: ON | OFF | YES | NO | TRUE | FALSE

Example:

```
net_service_name=
(description=
  (failover=on)
  (address =...)
  (address =...)
  )
(connect_data=(service_name=sales.com))
)
```

LOAD_BALANCE

Purpose:	When set to ON, instructs Net8 to progress through the list of listener addresses in a random sequence, balancing the load on the various listeners. When set to OFF, instructs Net8 to try the addresses sequentially until one succeeds. This parameter must be embedded under a DESCRIPTION_LIST, DESCRIPTION or ADDRESS_LIST.
Default:	ON for DESCRIPTION_LISTs
Values:	ON OFF YES NO TRUE FALSE
Example:	<pre>net_service_name= (description= (load_balance=on) (address =...) (address =...)) (connect_data=(service_name=sales.com))</pre>

SOURCE_ROUTE

Purpose:	When set to ON or YES, instructs Net8 use each address in order until the destination reached. This parameter is required when reaching the destination requires more than one address stop. This parameter is required for Oracle Connection Manager. For Oracle Connection Manager, an initial connection from the client to the Oracle Connection Manager is required, and a second connection from the Oracle Connection Manager to the listener is required. This parameter must be embedded under a DESCRIPTION_LIST, DESCRIPTION or ADDRESS_LIST.
Default:	OFF
Values:	YES, NO, ON, OFF
Example:	<pre>net_service_name= (description = (source_route=yes) (address=...) (address=...)) (connect_data=(sid=sales))</pre>

TYPE_OF_SERVICE

Purpose: Specifies the type of service to use for an Oracle RDB database. It is used by Rdb interface tools. This feature should only be used if the application supports both an Oracle Rdb and Oracle database, and you want the application to randomly choose (load balance). This parameter must be embedded under a DESCRIPTION.

Example:

```
net_service_name=
(description_list=
(description=
(address=...
)
(connect_data=
(service_name=generic)
(rdb_database=[.mf]mf_personal.rdb)
(global_name=alpha5))
(type_of_service=rdb_database))
(description=
(address=...
(connect_data=
(service_name=sales))
(type_of_service=oracle8_database)))
```

CONNECT_DATA Section

CONNECT_DATA

Purpose: Defines the database to which to connect. This parameter must be embedded under a DESCRIPTION.

Example:

```

net_service_name=
  (description=
    (address =...)
    (address =...)
  )
  (connect_data=(service_name=sales.com))

```

CONNECT_DATA offers the following parameters:

- [FAILOVER_MODE](#)
- [GLOBAL_NAME](#)
- [HS](#)
- [INSTANCE_NAME](#)
- [RDB_DATABASE](#)
- [SDU](#)
- [SERVER](#)
- [SERVICE_NAME](#)

FAILOVER_MODE

Purpose: Instructs Net8 to fail over to a different listener if the first listener fails during runtime. Depending upon the configuration, session or any SELECT statements which were in progress are automatically failed over. This parameter must be embedded under CONNECT_DATA.

This type of failover is called "Transparent Application Failover" and should not be confused with the "[Connect-Time Failover](#)" [FAILOVER](#) parameter. See *Oracle8i Tuning* for further information about the implementation of transparent application failover.

FAILOVER_MODE

Sub-Parameters

FAILOVER_MODE supports the following sub-parameters:

- **BACKUP** - Specifies the failover node.
- **TYPE - (Required)** Specifies the type of failover. Three types of Net8 failover functionality are available by default to Oracle Call Interface (OCI) applications:
 - SESSION**: Fails over the session; that is, if a user's connection is lost, a new session is automatically created for the user on the backup. This type of failover does not attempt to recover selects.
 - SELECT**: Allows users with open cursors to continue fetching on them after failure. However, this mode involves overhead on the client side in normal select operations.
 - NONE**: This is the default, in which no failover functionality is used. This can also be explicitly specified to prevent failover from happening.
- **METHOD**: Determines how fast failover occurs from the primary node to the backup node:
 - BASIC**: Establishes connections at failover time. This option requires almost no work on the backup server until failover time.
 - PRECONNECT**: Pre-establishes connections. This provides faster failover but requires that the backup instance be able to support all connections from every supported instance

FAILOVER_MODE

Example:

Shown below is an example of transparent application failover (FAILOVER_MODE) to a backup node called OPSHP2 during a failure of OPSHP1. Any SELECT statements in progress will be failed over.

```

net_service_name=
(description=
(address=
(protocol=tcp)
(host=opshp1)
(port=1521))
(connect_data=
(service_name=op.us.oracle.com)
(failover_mode=
(backup=opshp2)
(type=select)
(method=basic)
)
)
)

```

Shown below is transparent application failover (FAILOVER_MODE) being used with connect-time failover (FAILOVER). A connection will be tried first to OPSHP1, then to OPSHP2. If, after a connection, the node fails, the application will try to failover to the OPSHP1, then OPSHP2, reserving any SELECT statements in progress.

```

net_service_name=
(description=
(failover=on)
(address=
(protocol=tcp)
(host=opshp1)
(port=1521))
(address=
(protocol=tcp)
(host=opshp2)
(port=1521))
(connect_data=
(service_name=op.us.oracle.com)
(failover_mode=
(type=select)
)
)
)

```

GLOBAL_NAME

Purpose: Identifies the Oracle Rdb database.

Example:

```
net_service_name=  
(description=  
(address =...)  
(address =...)  
(connect_data=  
(service_name=generic)  
(rdb_database= [.mf]mf_personal.rdb)  
(global_name= alpha5)))
```

HS

Purpose: Instructs Net8 to connect to a non-Oracle system. This parameter must be embedded under CONNECT_DATA.

Default: None

Values: OK

Example:

```
net_service_name=  
(description =  
(address=...)  
(address=...)  
(connect_data=  
(sid=sales6)  
(hs=ok))
```

INSTANCE_NAME

Purpose: Identifies the database instance name to access. This parameter may be embedded under CONNECT_DATA. The instance name can be obtained from the INSTANCE_NAME parameter in the INITSID.ORA file.

The INSTANCE_NAME is typically the system identifier (SID) entered during installation or database creation.

Additional Information: "[Configuring Protocol Addresses](#)" on page 6-76 for information on setting the instance name string.

Example:

```
net_service_name=
(description =
  (address=...)
  (address=...)
(connect_data=
  (service_name=sales.com)
  (instance_name=sales1))
```

RDB_DATABASE

Purpose: Specifies the file name of an Oracle RDB database.

Example:

```
net_service_name=
(description=
  (address =...)
  (address =...)
(connect_data=
  (service_name=sales.com)
  (rdb_database= [.mf]mf_personal.rdb))
```

SDU

Purpose: Instructs Net8 to optimize the transfer rate of data packets being sent across the network with the session data unit (SDU) size you specify. This parameter may be embedded under CONNECT_DATA.

Default: 2048

Values: The SDU size should be set as a multiple of the normal transport frame size. Since the normal Ethernet frame size is 1024, the most efficient SDU size over an Ethernet protocol should be a multiple of 1024, but not more than four times the amount of 1024.

Example:

```
net_service_name=  
(description =  
  (address=...)  
  (address=...)  
(connect_data=  
  (server_name=sales.com)  
  (sdu=1024))
```

SERVER

Purpose: Instructs Net8 to use a dedicated server in an MTS environment. This parameter may be embedded under CONNECT_DATA.

Default: DEDICATED

Values: DEDICATED | SHARED

Example:

```
net_service_name=  
(description =  
  (address=...)  
  (address=...)  
(connect_data=  
  (server_name=sales.com)  
  (server=dedicated))
```

SERVICE_NAME

Purpose: Identifies the release 8.1 service to access. This parameter must be embedded under CONNECT_DATA. The SERVICE_NAME is typically set to the global database name, a name comprised of the database name and domain name, entered during installation or database creation.

Additional Information: See "[Setting Service Names and Instance Names](#)" on page 6-3 for information on setting the instance name.

Example:

```
net_service_name=  
(description =  
  (address=...)  
  (address=...)  
(connect_data=  
  (service_name=sales.com)  
  (instance_name=sales1))
```

SID

Purpose: Identifies the database instance for pre-release 8.1 databases. This parameter must be embedded under CONNECT_DATA.

Example:

```
net_service_name=  
(description =  
  (address=...)  
  (address=...)  
(connect_data=  
  (sid=sales))
```

Listener Parameters (LISTENER.ORA)

The following items are available in the network listener configuration file (LISTENER.ORA).

- [Listener Address Section](#)
- [SID_LIST_listener_name Static Service Section](#)
- [Control Parameters](#)

LISTENER.ORA is located in `$ORACLE_HOME/network/admin` on UNIX and `ORACLE_HOME\network\admin` on Windows NT.

Listener Address Section

The listener address section of the LISTENER.ORA defines the protocol address(es) of the listener.

DESCRIPTION

Purpose:

Defines a listener protocol address, as well as the [presentation](#) and [session](#) communication layers to use.

Prior to this release, Net8 only supported connections that used the Two-Task Common (TTC) presentation layer and Transparent Network Substrate (TNS) Network session (NS) layer to establish client connections.

In order for clients to connect to the Java option in the database, the presentation layer and session layer support have been expanded. For those clients wishing to connect to the Java option, the General Inter-Orb Protocol (GIOP) presentation layer is used with no session layer.

This parameter may be embedded under a DESCRIPTION_LIST.

It is not necessary to use DESCRIPTIONs if all the protocol addresses specified use TTC and NS.

Additional Information: See "[Configuring Java Option Connections](#)" on page 7-50 for configuration information.

Example:

```
listener_name =
  (description_list =
    (description =
      (address=
        (protocol=ipc)
        (key=extproc0))
      )
    (description =
      (protocol_stack =
        (presentation=giop)
        (session=raw))
      (address=
        (protocol=tcp)
        (host=sales-pc)
        (port=2481))
      )
    )
  )
```

DESCRIPTION_LIST

Purpose: Defines a list of listener protocol address and protocol stack descriptions.

Example:

```
listener_name =
  (description_list =
    (description =
      (address=
        (protocol=ipc)
        (key=extproc0))
      )
    (description =
      (protocol_stack =
        (presentation=giop)
        (session=raw))
      (address=
        (protocol=tcp)
        (host=sales-pc)
        (port=2481))
      )
    )
  )
```

ADDRESS

Purpose: Defines a listener protocol address. This parameter may be embedded under a ADDRESS_LIST or DESCRIPTION.

Additional Information: See "[Configuring Protocol Addresses](#)" on page 6-76 for descriptions of the correct parameters to use for each protocol.

Example:

```
listener_name =
  (address=
    (protocol=ipc)
    (key=extproc0))
  (address=
    (protocol=tcp)
    (host=sales-pc)
    (port=2481))
```

ADDRESS_LIST

Purpose: Defines lists of listener protocol addresses. Because there will only be one list of addresses, it is *not* necessary to use this parameter. By default operation, the Net8 Assistant and Net8 Configuration Assistant create an ADDRESS_LIST each time an address is specified, even though there is only one list. This parameter may be embedded under a DESCRIPTION or DESCRIPTION_LIST.

Example:

```
listener_name =
  (description_list =
    (description =
      (address_list =
        (address=
          (protocol=ipc)
          (key=extproc0))
        )
      )
    (description =
      (protocol_stack =
        (presentation=giop)
        (session=raw)
      )
      (address_list =
        (address=
          (protocol=tcp)
          (host=sales-pc)
          (port=2481))
        )
      )
    (description =
      (address_list =
        (address=
          (protocol=tcp)
          (host=sales-pc)
          (port=1521))
        )
      )
    )
  )
)
```

PROTOCOL_STACK

Purpose: Prior to this release, Net8 only supported connections that used the Two-Task Common (TTC) presentation layer and Network Session (NS) layer to establish client connections. For those clients wishing to connect to the Java option, the General Inter-Orb Protocol (GIOP) presentation layer is used with no session layer. This parameter must be embedded under a DESCRIPTION. It is not necessary to use PROTOCOL_STACK if the protocol address(es) specified use TTC and NS.

Additional Information: See "[Configuring Java Option Connections](#)" on page 7-50 for further configuration information.

Example:

```
listener_name =
  (description_list =
    (description =
      (address=
        (protocol=ipc)
        (key=extproc0))
      )
    (description =
      (protocol_stack =
        (presentation=giop)
        (session=raw))
      (address=
        (protocol=tcp)
        (host=sales-pc)
        (port=2481))
      )
    )
  )
```

PRESENTATION

Purpose: Identifies the presentation layer.

Default: TTC

Values: TTC | GIOP

Typical Net8 clients use a [presentation](#) layer of [Two-Task Common](#) (TTC).

IIOIP clients require a presentation layer of GIOP. GIOP is an alias for *either* oracle.aurora.server.SGiopServer or oracle.aurora.server.GiopServer using TCP/IP or TCP/IP with SSL.

Additional Information: See "[Configuring Java Option Connections](#)" on page 7-50 for further configuration information.

Example:

```
listener_name =
  (description_list =
    (description =
      (address=
        (protocol=ipc)
        (key=extproc0))
      )
    (description =
      (protocol_stack =
        (presentation=giop)
        (session=raw))
      (address=
        (protocol=tcp)
        (host=sales-pc)
        (port=2481))
      )
    )
  )
```

SESSION

Purpose: Identifies the session layer.

Default: NS

Values: NS | RAW

IIOP clients require no [session](#) layer (RAW).

Typical Net8 clients use a session layer of [Transparent Network Substrate \(TNS\) Network Session \(NS\)](#).

Example:

```
listener_name =
  (description_list =
    (description =
      (address=
        (protocol=ipc)
        (key=extproc0))
      )
    (description =
      (protocol_stack =
        (presentation=giop)
        (session=raw))
      (address=
        (protocol=tcp)
        (host=sales-pc)
        (port=2481))
      )
    )
  )
```

SID_LIST_*listener_name* Static Service Section

The SID_LIST section of the LISTENER.ORA defines the system identifier (SID) of the database served by the listener. This section is only valid for version 7.x and version 8.0 Oracle databases, as instance information is now automatically registered with the listener. Static configuration is also required for other services, such as external procedures and heterogeneous services, and some management tools, including Oracle Enterprise Manager.

CAUTION: Implementing connect-time failover does not allow use of static service configuration parameters in the LISTENER.ORA file. However, static configuration is required for Oracle8i release 8.0 or Oracle7 databases and Oracle Enterprise Manager. For further information about statically configuring the listener, see ["Configuring a Listener with Service Information"](#) on page 6-71.

See the *Oracle8i Parallel Server Setup and Configuration Guide* for instructions on setting up connect-time failover in an Oracle Enterprise environment.

```
sid_list_listener_name=
  (sid_list=
    (sid_desc=
      (global_dbname=global_database_name)
      (sid_name=sid)
      (oracle_home=oracle_home)
      (prespawn_max=99)
      (prespawn_list=
        (prespawn_desc=
          (protocol=tcp)
          (pool_size=10)
          (timeout=2))
        (prespawn_desc=...))
      (sid_desc=...))
  )
```

SID_LIST contains the following parameters:

GLOBAL_DBNAME

Purpose: The global database name is the name and domain name of the database, as given in SERVICE_NAMES parameter in the database initialization parameter file (INITSID.ORA). If you want to refer to the database by its global database name on the network, then you must specify that global database name to the listener. This parameter must be embedded under SID_DESC and should match the value of the SERVICE_NAMES parameter.

Example:

```
sid_list_listener_name=  
  (sid_list=  
    (sid_desc=  
      (global_dbname=oracle.com)  
      (sid_name=orcl)  
      (oracle_home=/usr/oracle)  
    )  
  )  
)
```

ORACLE_HOME

Purpose: Identifies the Oracle home location of the database that you are specifying. This parameter must be embedded under SID_DESC.

Example:

```
sid_list_listener_name=  
  (sid_list=  
    (sid_desc=  
      (sid_name=extproc)  
      (oracle_home=/usr/oracle)  
      (program=extproc)  
    )  
  )  
)
```

PROGRAM

Purpose: Identifies the service's executable program name. For example, extproc.exe.

Example:

```
sid_list_listener_name=  
  (sid_list=  
    (sid_desc=  
      (sid_name=extproc)  
      (oracle_home=oracle)  
      (program=extproc)  
    )  
  )  
)
```

SID_NAME

Purpose: Identifies the SID of the database. Its value is operating system specific. This parameter must be embedded under SID_DESC.

This parameter should match the value of the INSTANCE_NAME parameter in the INITSID.ORA file.

Example:

```
sid_list_listener_name=  
  (sid_list=  
    (sid_desc=  
      (global_dbname=oracle.com)  
      (sid_name=orcl)  
      (oracle_home=/usr/oracle)  
    )  
  )  
)
```

SID_LIST

Purpose: Identifies a list of SID descriptions.

Example:

```
sid_list_listener_name=
  (sid_list=
    (sid_desc=...)
    (sid_desc=...)
  )
```

SID_DESC

Purpose: Defines a SID. This parameter may be embedded under SID_LIST.

Example:

```
sid_list_listener_name=
  (sid_list=
    (sid_desc=...)
    (sid_desc=...)
  )
```

PRESPAWN_MAX

Purpose: Specifies the maximum number of prespawned dedicated server processes the listener will create. This number must be at least as many as the sum of the pool size for each protocol. Set this value to a large number so that prespawned dedicated server processes are always available for new connections. This parameter must be embedded under SID_LIST.

Prespawned dedicated servers cannot be configured on Windows NT.

Example:

```
sid_list_listener_name=
  (sid_list=
    (sid_desc=
      (global_dbname=oracle.com)
      (sid_name=sid)
      (oracle_home=/usr/oracle)
      (prespawn_max=99)
    )
  )
```

PRESPAWN_LIST

Purpose: Specifies a list of prespawnd dedicated server process descriptions. This parameter must be embedded under SID_DESC.

Example:

```

sid_list_listener_name=
  (sid_list=
    (sid_desc=
      (global_dbname=oracle.com)
      (sid_name=sid)
      (oracle_home=/usr/oracle)
      (prespawn_max=99)
      (prespawn_list=
        (prespawn_desc=
          (protocol=tcp)
          (pool_size=10)
          (timeout=2))
          (prespawn_desc=...)))
    )
  )

```

PRESPAWN_DESC

Purpose: Defines the protocol on which the listener creates prespawnd dedicated server processes, as well as characteristics of the prespawnd dedicated server. Each protocol must have its own PRESPAWN_DESC description. This parameter must be embedded under PRESPAWN_LIST.

Additional Information: See ["Configuring Prestarted or Prespawnd Dedicated Servers"](#) on page 6-73 for descriptions of PROTOCOL, POOL_SIZE, and TIMEOUT.

Example:

```

sid_list_listener_name=
  (sid_list=
    (sid_desc=
      (global_dbname=oracle.com)
      (sid_name=sid)
      (oracle_home=/usr/oracle)
      (prespawn_max=99)
      (prespawn_list=
        (prespawn_desc=
          (protocol=tcp)
          (pool_size=10)
          (timeout=2))
          (prespawn_desc=...)))
    )
  )

```

Control Parameters

The following parameters control the behavior of the listener:

CONNECT_TIMEOUT_listener_name

Purpose: Determines the amount of time the listener will wait for a valid connection request after a connection has been started.

Default: 10 seconds

Example: connect_timeout_listener=12

LOG_DIRECTORY_listener_name

Purpose: Controls the destination directory of the log file.

Default: \$ORACLE_HOME/network/log on UNIX and \$ORACLE_HOME/network/log on Windows NT.

Example: log_directory_listener=/oracle/network/admin/log

LOG_FILE_listener_name

Purpose: Sets the name of the log file for the listener.

Default: LISTENER.LOG

Example: log_file_listener=list.log

PASSWORDS_listener_name

Purpose: Sets a non-encrypted password for authentication to the Listener Control Utility (LSNRCTL). Allows one or more passwords. If this optional parameter specifies one or more passwords, then the use of one of these passwords is required to perform DBA tasks against the network listener using the LSNRCTL.

Default: oracle

Example: passwords_listener=(oracle8)

STARTUP_WAITTIME_listener_name

Purpose: Sets the number of seconds that the network listener sleeps before responding to the first listener control status command.

Default: 0 seconds

Example: startup_waittime_listener=5

TRACE_DIRECTORY_listener_name

Purpose: Controls the destination directory of the log file.

Default: \$ORACLE_HOME/network/trace on UNIX and \$ORACLE_HOME/network/trace on Windows NT.

Example: trace_directory_listener=/oracle/network/admin/trace

TRACE_FILE_listener_name

Purpose: Sets the name of the trace file for the listener.

Default: LISTENER.TRC

Example: trace_file_listener=list.trc

TRACE_LEVEL_listener_name

Purpose: Turns tracing on/off to a certain specified level.

Default: OFF

Values

- OFF - No trace output
- USER - User trace information
- ADMIN - Administration trace information
- SUPPORT - WorldWide Customer Support trace information

Example: `trace_level_listener=admin`

USE_PLUG_AND_PLAY_listener_name

Purpose: Instructs the listener to find and register database information with an Oracle Names Server.

When a client requests a connection to a server, the Oracle Names server retrieves the necessary information, such as the service name and SID, from the listener.

Default: OFF

Values: ON | OFF

Example: `use_plug_and_play=on`

Oracle Names Parameters (NAMES.ORA)

The following parameters are available in an Oracle Names configuration file (NAMES.ORA). NAMES.ORA is located in \$ORACLE_HOME/network/admin on UNIX and *ORACLE_HOME*\network\admin on Windows NT.

NAMES.ADDRESSES

Purpose:	Describes the address on which the Oracle Names server listens. Any valid ADDRESS or ADDRESS_LIST is allowed. Additional Information: See: <ul style="list-style-type: none">▪ "ADDRESSES and ADDRESS_LISTS" on page C-5 for description of address syntax▪ "Configuring Protocol Addresses" on page 6-76 for descriptions of the correct parameters to use for each protocol
Default:	names.addresses= (address=(protocol=tcp)(host=oranamesrvr0)(port=1575)

NAMES.ADMIN_REGION

Purpose: Describes the data source for an administrative region. If set, this parameter defines a database as a repository for information. If this parameter is not set, replication of data between Oracle Names server caches.

Default: NULL

Example:

```
names.admin_region=  
  (region=(name=local_region)  
  (type=rosdb)  
  (userid=names)  
  (password=names)  
  (description=  
    (address=(protocol=tcp)  
    (host=nineva)(port=1575)  
  )  
)  
(connect_data=  
  (service_name=em)  
)  
)  
(version=34619392)  
(refresh=14400)  
(retry=600)  
(expire=259200)
```

NAMES.AUTHORITY_REQUIRED

Purpose: Determines whether system queries require authoritative answers.

Default: FALSE

Example: names.authority_required=true

NAMES.CONNECT_TIMEOUT

Purpose:	Limits the amounts of time the Oracle Names server will wait for the connection from a client to complete.
Default:	3 seconds
Minimum Value:	1 second
Maximum Value:	600 seconds
Example:	<code>names.connect_timeout=8</code>

NAMES.AUTO_REFRESH_EXPIRE

Purpose:	Specifies the amount of time in seconds the Oracle Names server caches other region's database server addresses which have been obtained through NAMES.DOMAIN.HINTS. At the end of this interval, the Oracle Names server issues a query to the other region database servers to refresh the address.
Default:	600 seconds
Acceptable Values:	60-1209600 seconds
Example:	<code>names.auto_refresh_expire=1200000</code>

NAMES.AUTO_REFRESH_RETRY

Purpose:	Specifies the interval in seconds that the Oracle Names server will retry when the auto refresh query fails.
Default:	180
Minimum Value:	60
Maximum Value:	3600
Example:	<code>names.auto_refresh_retry=180</code>

NAMES.CACHE_CHECKPOINT_FILE

Purpose: Specifies the name and path of the file to which the Oracle Names server writes its checkpoint file.

Default: `$ORACLE_HOME/network/names/ckpcch.ora` on **UNIX** and `ORACLE_HOME\network\names\ckpcch.ora` on Windows NT.

Example: `names.cache_checkpoint_
file=c:\oracle\network\names\ccheck.ora`

NAMES.CACHE_CHECKPOINT_INTERVAL

Purpose: Indicates the interval at which a Oracle Names server writes a checkpoint of its stored data to the checkpoint file. Each Oracle Names server can periodically write its cached data to a file to protect against startup failures.

Default: 0 (disabled)

Minimum Value: 10

Maximum Value: 259200 (3 days)

Example: `names.cache_checkpoint_interval=24`

NAMES.CONFIG_CHECKPOINT_FILE

Purpose: Specifies the name and path of the file used to checkpoint Oracle Names server configuration settings.

Default: `$ORACLE_HOME/network/names/ckcfg.ora` on **UNIX** and `ORACLE_HOME\network\names\ckcfg.ora` on Windows NT.

Example: `names.config_checkpoint_
file=c:\oracle\network\names\configck.ora`

NAMES.DEFAULT_FORWARDERS

Purpose: Address list of other Oracle Names servers which are used to forward queries.

Example:

```
names.default_forwarders= (forwarder_list=
  (forwarder=(name=rootserv1.com)(address=(protocol=tcp)
    (port=4200)(host=roothost))))
```

NAMES.DEFAULT_FORWARDERS_ONLY

Purpose: When TRUE the Oracle Names server forwards queries only to those Oracle Names servers listed as default forwarders.

Default: FALSE

Usage Notes: If set to FALSE, Oracle Names servers listed as default forwarders will be called before Oracle Names servers found in the cache.

NAMES.DOMAIN_HINTS

Purpose: Lists the names, addresses and domains of all servers in one or more foreign regions. Enables the Oracle Names server to know about other regions' servers. This includes at least the root region for all servers who are not in the root region. Other regions may be provided as optimization requires.

Example:

```
names.domain_hints=
  (hint_desc=(hint_list=
    (hint=(name=rootserv1.com)
      (address=(protocol=tcp)(host=nineva)(port=4200)
    )
  )
)
```

NAMES.DOMAINS

Purpose: List of domains in the server's local region, as well as the default time to live (TTL) for data in those domains.

Example:

```
names.domains=
(domain_list=
(domain=(name=)(min_ttl=86400))
(domain=(name=com)(min_ttl=86400))
)
```

NAMES.FORWARDING_AVAILABLE

Purpose: The Oracle Names server forwards operations to foreign regions if set to ON.

Default: ON

Values: ON | OFF

Example: `names.forwarding_available=off`

NAMES.FORWARDING_DESIRED

Purpose: If set to TRUE, then this Oracle Names servers forwards queries.

Default: TRUE

Values: TRUE | FALSE

Example: `names.forwarding_desired=true`

NAMES.KEEP_DB_OPEN

Purpose: Specifies whether or not to attempt to keep the TNS connection to the region database open between operations. If set to FALSE, the connection will be closed after each load, reload or reload-check.

Default: TRUE

Values: TRUE | FALSE

Example: `names.keep_db_open=FALSE`

NAMES.LOG_DIRECTORY

Purpose:	Indicates the name of the directory where the log file for Oracle Names server operational events are written.
Default:	<code>\$ORACLE_HOME/network/log</code> on UNIX and <code>ORACLE_HOME\network\log</code> on Windows NT.
Example:	<code>names.log_directory=c:\oracle\network\names</code>

NAMES.LOG_FILE

Purpose:	Indicates the name of the output file to which Oracle Names server operational events are written. The file name extension is always <code>.LOG</code> . Do not enter an extension for this parameter.
Default:	<code>NAMES</code>
Example:	<code>names.log_file=onames</code>

NAMES.LOG_STATS_INTERVAL

Purpose:	Specifies the number of seconds between full statistical dumps in the log file.
Default:	<code>0 (0=OFF)</code>
Minimum Value:	<code>10</code> (least possible ON value)
Maximum Value:	<code>none</code>
Example:	<code>names.log_stats_interval=12</code>

NAMES.LOG_UNIQUE

Purpose:	If set to <code>TRUE</code> , then the log file name will be unique and not overwrite existing log files.
Default:	<code>FALSE</code>
Values:	<code>TRUE FALSE</code>
Example:	<code>names.log_unique=true</code>

NAMES.MAX_OPEN_CONNECTIONS

Purpose:	Specifies the number of connections that the Oracle Names server can have open at any given time. The value is generated as the value 10 or the sum of one connection for listening, five for clients, plus one for each foreign domain defined in the local administrative region, whichever is greater. The calculated value is acceptable for most installations.
Default:	Calculated based on entered data.
Minimum Value:	2
Maximum Value:	64
Example:	<code>names.max_open_connections=52</code>

NAMES.MAX_REFORWARDS

Purpose:	Specifies the maximum number of times the server attempts to forward an operation.
Default:	2
Minimum Value:	1
Maximum Value:	15
Example:	<code>names.max_reforwards=2</code>

NAMES.MESSAGE_POOL_START_SIZE

Purpose:	Determines the initial number of messages allocated in the server's message pool which are used for incoming or outgoing forwarded messages.
Default:	10
Minimum Value:	3
Maximum Value:	256
Example:	<code>names.message_pool_start_size=10</code>

NAMES.NO_MODIFY_REQUESTS

Purpose:	If set to TRUE, the server refuses any operations which modify the data in its region.
Default:	FALSE
Values:	TRUE FALSE
Example:	<code>names.no_modify_requests=true</code>

NAMES.NO_REGION_DATABASE

Purpose:	If set to TRUE, the server does not look for a region database.
Default:	FALSE
Example:	<code>names.no_region_database=true</code>

NAMES.PASSWORD

Purpose:	<p>Sets the password for an Oracle Names server, so that certain sensitive operations, such as STOP, RESTART and RELOAD operations in NAMESCTL are secure.</p> <p>If this parameter is set with the Net8 Assistant, the password is encrypted. A clear-text password may be made manually. If the password is clear-text, ensure the NAMESCTL.INTERNAL_ENCRYPT_PASSWORD parameter is set in the SQLNET.ORA file is set to FALSE.</p>
Default:	None
Example:	<code>names.password=625926683431aa55</code>

NAMES.REGION_CHECKPOINT_FILE

Purpose:	Specifies the name and path of the file used to checkpoint region data (for example, domain addresses, database addresses of Oracle Names servers in the local region)
Default:	<code>\$ORACLE_HOME/network/names/ckpreg.ora</code> on UNIX and <code>ORACLE_HOME\network\names\ckpreg.ora</code> on Windows NT.
Example:	<code>names.region_checkpoint_file=c:\oracle\network\names\regionck.ora</code>

NAMES.RESET_STATS_INTERVAL

Purpose:	Specifies the number of seconds during which the statistics collected by the Oracle Names servers should accumulate. At the frequency specified, they are reset to zero. The default value of 0 means never reset statistics.
Default:	0 (never reset)
Minimum Value:	10
Maximum Value:	none
Example:	<code>names.reset_stats_interval=15</code>

NAMES.SAVE_CONFIG_ON_STOP

Purpose:	If set to TRUE, the server will save its runtime configuration settings back into NAMES.ORA. Any parameters which were modified through NAMESCTL SET operations will replace prior NAMES.ORA settings.
Default:	FALSE
Example:	<code>names.save_config_on_stop=FALSE</code>

NAMES.SERVER_NAME

Purpose:	Each Oracle Names server is uniquely identified by a name. All configuration references to a particular Oracle Names server use this name.
Default:	<code>ONAMES_onames_server</code>
Example:	<code>names.server_name=namesrv1.us.oracle.com</code>

NAMES.TRACE_DIRECTORY

Purpose:	Indicates the name of the directory to which trace files from a Oracle Names server trace session are written.
Default:	<code>\$ORACLE_HOME/network/trace</code> on UNIX and <code>ORACLE_HOME/network/trace</code> on Windows NT.
Example:	<code>names.trace_directory=/oracle/network/admin/trace</code>

NAMES.TRACE_FILE

Purpose:	Indicates the name of the output file from a Oracle Names server trace session. The file name extension is always .TRC.
Default:	NAMES
Example:	<code>names.trace_file=onames</code>

NAMES.TRACE_FUNC

Purpose:	Enables internal mechanism to control tracing by function name.
Default:	FALSE
Example:	<code>names.trace_func=false</code>

NAMES.TRACE_LEVEL

Purpose:	Indicates the level at which the Oracle Names server is to be traced.
Default:	OFF
Values	<ul style="list-style-type: none">■ OFF - No trace output■ USER - User trace information■ ADMIN - Administration trace information■ SUPPORT - WorldWide Customer Support trace information
Example:	<code>names.trace_level=admin</code>

NAMES.TRACE_UNIQUE

Purpose:	Indicates whether each trace file has a unique name, allowing multiple trace files to coexist. If the value is set to ON, a process identifier is appended to the name of each trace file generated.
Default:	ON
Values	ON OFF
Example:	<code>names.trace_unique=on</code>

Oracle Connection Manager Parameters (CMAN.ORA)

The following parameters are available in an Oracle Connection Manager configuration file (CMAN.ORA). CMAN.ORA is located in \$ORACLE_HOME/network/admin on UNIX and ORACLE_HOME\network\admin on Windows NT.

CMAN

Purpose:	Specifies listening addresses for Oracle Connection Manager gateway process, CMGW .
Default:	<code>cmn=(address=(protocol=tcp)(host=local_host)(port=1630))</code>
Syntax:	<code>cmn= [(address_list=] (address= ...) [(address= ...)]</code>

CMAN_ADMIN

Purpose:	Specifies listening addresses for Oracle Connection Manager administrative process, CMADMIN .
Default:	<code>cmn_admin=(address=(protocol=tcp)(host=anyhost)(port=1830))</code>
Syntax:	<code>[(address_list=] (address= ...) [(address= ...)]</code>

CMAN_PROFILE

Purpose:	Sets parameters related to the running of Oracle Connection Manager.
Default:	<ul style="list-style-type: none"> ■ ANSWER_TIMEOUT=0 ■ AUTHENTICATION_LEVEL=0 ■ LOG_LEVEL=0 ■ MAX_FREELIST_BUFFERS=2048 ■ MAXIMUM_CONNECT_DATA=1024 ■ MAXIMUM_RELAYS=128 ■ RELAY_STATISTICS=NO ■ SHOW_TNS_INFO=NO ■ TRACE_DIRECTORY=\$ORACLE_HOME/network/trace on UNIX and <i>ORACLE_HOME</i>\network\trace on Windows NT ■ TRACING=NO ■ USE_ASYNC_CALL=YES
Values:	<ul style="list-style-type: none"> ■ ANSWER_TIMEOUT=[0 to <i>n</i>] ■ AUTHENTICATION_LEVEL=[0, 1] ■ LOG_LEVEL=[0-4] ■ MAXIMUM_CONNECT_DATA=[257-4096] ■ MAX_FREELIST_BUFFERS=[0-10240] ■ MAXIMUM_RELAYS=[0-2048] ■ RELAY_STATISTICS=[YES,NO] ■ SHOW_TNS_INFO=[YES, NO] ■ TRACE_DIRECTORY=<i>pathname</i> ■ TRACING=[YES, NO] ■ USE_ASYNC_CALL=[YES, NO]

CMAN_PROFILE

Example:

```

cmn_profile =
(parameter_list=
(maximum_relays=512)
(log_level=1)
(tracing=yes)
(relay_statistics=yes)
(show_tns_info=yes)
(use_async_call=yes)
(authentication_level=0)
)

```

CMAN_PROFILE Parameter	Description
ANSWER_TIMEOUT	The number of seconds that Oracle Connection Manager will use to time out the protocol handshake associated with an incoming connection request. The range is 0 to <i>n</i> .
AUTHENTICATION_LEVEL	1 instructs the CMAN to reject connect requests that are not using Secure Network Services. Secure Network Services is part of the Oracle Advanced Security option. 0 is the default, which means Secure Network Services is not required
LOG_LEVEL	Determines the level of logging performed by the Oracle Connection Manager. <ul style="list-style-type: none"> ■ default is 0, which means no logging is performed ■ range is 0 to 4: <ul style="list-style-type: none"> level 1 - basic reporting level 2 - RULE_LIST matching lookup reporting level 3 - relay blocking reporting level 4 - relay I/O counts reporting <p>The CMGW gateway process creates a log file called <i>CMAN_PID.LOG</i>, and the CMADMIN administrative process creates a log file called <i>CMADM_PID.LOG</i>. The log files are located in <code>\$ORACLE_HOME/network/log</code> on UNIX and <code>ORACLE_HOME\network\log</code> on Windows NT.</p>

CMAN_PROFILE Parameter	Description
MAX_FREELIST_BUFFERS	Determines the maximum number of buffers that TNS will keep in freelist for later user instead of returning them to the operating system after a relay gets closed. The range is 0 to 10240.
MAXIMUM_CONNECT_DATA	Limits the connect data string length of the incoming connection requests. The range is 257-4096.
MAXIMUM_RELAYS	Determines the maximum number of concurrent connections Oracle Connection Manager will support. The range is 0 to 2048.
RELAY_STATISTICS	YES instructs the CMGW process to maintain statistics pertaining to relay I/O activities, such as: <ul style="list-style-type: none"> ▪ number of IN bytes ▪ number of OUT bytes ▪ number of IN packets ▪ number of OUT packets This information is stored in the CMAN_PID.LOG file.
SHOW_TNS_INFO	YES instructs the CMGW process to include TNS events in the CMAN_PID.LOG file.
TRACING	YES enables tracing for the Oracle Connection Manager. The CMGW gateway process creates a trace file called CMAN_PID.TRC, and the CMADMIN administrative process creates a trace file called CMADM_PID.TRC.
TRACE_DIRECTORY	Specifies the trace directory.
USE_ASYNC_CALL	YES instructs the CMGW process to use all asynchronous functions while in the answering or calling phase of establishing a Net8 connection. Note: CMGW supports out-of-band breaks, it will forward it on to the server.

CMAN_RULES

Purpose: Sets the rules for the network access control portion of Oracle Connection Manager.

Syntax:

```
cmn_rules=
  (rules_list=
    (rule=(src=hostname)(dst=hostname)(srv= service_name|sid)
      (act=accept))
    [(rule= ...)]
  )
```

Values:

- SRC - Source host name or IP address (in dot notation) of session request
- DST - Destination server host name or IP address (in dot notation)
- SRV - Database server SID
- ACT - Accept or Reject incoming requests with the previous characteristics.

Usage Notes

The wildcard for host name is the single character 'x'. In the case of an IP address (d.d.d.d), you may wild card the individual d's with an 'x'.

Note: If client's connect string contain both SID and SERVICE_NAME, then both names requested need to be permitted by the rules respectively in order for the client's access to be allowed.

Example:

```
cmn_rules=
  (rule_list=
    (rule=(src=client1-pc)(dst=sales-pc)(srv=sales.com)(act=reject))
    (rule=(src=144.25.23.45)(dst=144.25.187.200)(srv=db1)(act=accept))
    (rule=(src=spcstn1)(dst=db2-pc)(srv=db2)(act=accept))
  )
```

Protocol-Specific Parameters (PROTOCOL.ORA)

The following parameters in the PROTOCOL.ORA configuration file are applicable to Net8.

protocol.EXCLUDED_NODES

Purpose:	Defines which nodes do not have validnode checking.
Example:	<code>protocol.excluded_nodes= (hostname ip_address, hostname ip_address, ...)</code>
Example:	<code>tcp.excluded_nodes= (hr.com, 144.25.5.25)</code>

protocol.INVITED_NODES

Purpose:	Defines which nodes have validnode checking. This list takes precedence over the <i>protocol.EXCLUDED_NODES</i> if both lists are present.
Syntax:	<code>protocol.invited_nodes= (hostname ip_address, hostname ip_address, ...)</code>
Example:	<code>tcp.invited_nodes= (sales.com, 144.185.5.73)</code>

protocol.VALIDNODE_CHECKING

Purpose:	Restricts connection access of clients to destinations with enabling host privilege.
Default:	NO
Values:	YES NO
Example:	<code>tcp.validnode_checking=yes</code>

TCP.NODELAY

Purpose:	Specifies no delays in buffer flushing.
Default:	NO
Values:	YES NO
Example:	<code>tcp.nodelay=yes</code>

Glossary

A

access control

See [Net8 access control](#).

address

A unique network location used to identify a network object, such as a listener address, Oracle Connection Manager, or Oracle Names server. Addresses have a specific format and must be unique. See also [well-known Oracle Names server](#).

administrative region

An organizational entity for administering Net8 network components. Each administrative region includes:

- one or more domains
- one or more Oracle Names servers
- one or more databases and listeners

alias

An alternative name for an existing network object. Once an alias is created, it is resolved to the same name as the initial network object. An Oracle Names server stores aliases for any defined net service name, database server or database link.

API

See [Net8 Open](#).

ASCII character set

Stands for American Standard Code for Information Interchange character set, a convention for representing alphanumeric information using digital data. The collation sequence used by most computers with the exception of IBM and IBM-compatible computers. Contrast with EBCDIC character set.

B**bequeath**

If the listener and server exist on the *same* node, the listener may create or spawn dedicated servers as connect requests are received. Dedicated servers are committed to one network session only and exist for the duration of that network session. The sequence of events that occur when the listener creates a dedicated server process and passes or "bequeaths" control of a network session to it.

C**cache**

Memory that stores recently-accessed data so that subsequent requests to access the same data can be processed quickly.

CDS

Cell Directory Service

central administration

An Oracle Names network where network management consists of one *administrative region* for the entire network. With central administration, all Oracle Names servers know about one another and about all the services in the network. Contrast with *delegated administration*.

CMAN.ORA file

A configuration file that specifies one or more listeners address for the Oracle Connection Manager, Oracle Connection Manager parameters, and *Net8 access control* rules.

Oracle Connection Manager Control Utility (CMCTL)

A utility included with Net8 to control various functions, such as to start, stop, and get the status of the Oracle Connection Manager.

client

A user, software application, or computer that requests the services, data, or processing of another application or computer. In a two-task environment, the client is the user process. In a network environment, the client is the local user process and the server may be local or remote.

client load balancing

Load balancing, whereby a client can randomly choose between the listeners for its connect requests if more than one listener services a single database. This randomization allows all listeners to share the burden of servicing incoming connect requests.

client profile

The properties of a client, which may include the preferred order of *naming methods*, client and server *logging* and *tracing*, the domain from which to request names, and other client options for Oracle Names and Oracle Advanced Networking Option.

client-server architecture

Software architecture based on a separation of processing between two CPUs, one acting as the client in the transaction, requesting and receiving services, and the other as the server that provides services in a transaction.

configuration files

Files that are used to identify and characterize the components of a network. Configuration is largely a process of naming network components and identifying relationships among those components.

CONNECT DATA

A portion of the connect descriptor, introduced by the keyword CONNECT DATA, that specifies the service to which the connection is to be made.

connect descriptor

A specially formatted description of the destination for a network connection in the *TNSNAMES.ORA file*. A connect descriptor contains the:

- network route to the service, including the location of the listener through a protocol address
- *service name*(s) used to identify the release 8.1 services, or *Oracle System Identifier (SID)* used to identify a pre-release 8.1 databases
- optional instance name of the release 8.1 service to which to connect

Clients and servers use this net service name when making a connection with an application.

connect-time failover

A client connect request is forwarded to a another listener if first listener is not responding. Connect-time failover is enabled by *database instance registration*, because the listener knows if an instance is up prior to attempting a connection.

connection

An interaction between two processes on a network. Connections are originated by an initiator (client), who requests a connection with a destination (server).

connection concentration

A feature of Oracle Connection Manager that consolidates multiple connection requests from clients to establish a single connection to a server in order to conserve server resources.

connection load balancing

Load balancing, whereby the number of active connections among various instances and dispatchers for the same service are balanced. This enables listeners to make their routing decisions based on how many connections each dispatcher has *and* on how loaded the nodes that the instances run.

connection pooling

A resource utilization and user scalability feature that allows you to maximize the number of physical network connections to a multi-threaded server. A feature of Oracle Connection Manager that consolidates multiple connection requests from clients to establish a single connection to a server in order to conserve server resources.

connection request

A notification sent by an initiator and received by a listener that indicates that the initiator wants to start a connection.

connect string

Information the user passes to a service to connect, such as user name, password and *net service name*. For example:

connect *username/password@net_service_name*

D**data packet**

See *packet*.

database administrator (DBA)

(1) A person responsible for operating and maintaining an Oracle Server or a database application. (2) An Oracle username that has been given DBA privileges and can perform database administration functions. Usually the two meanings coincide. Many sites have multiple DBAs.

database initialization file

See *initialization file (INITSID.ORA)*.

database instance registration

Database instances register themselves with the listener when started. Database instance registration is comprised of two elements:

- service registration provides the listener with instance information, such as database service names and instance names.
- MTS dispatcher registration provides dispatcher information to the listener

database link

A network object stored in the local database or in the network definition that identifies a remote database, a communication path to that database, and optionally, a username and password. Once defined, the database link is used to access the remote database.

A public or private database link from one database to another is created on the local database by a DBA or user.

A global database link is created automatically from each database to every other database in a network with Oracle Names. Global database links are stored in the network definition.

See also [global database link](#), [private database link](#), and [public database link](#).

decentralized administration

See [delegated administration](#).

dedicated server

A server that requires its own dedicated server process for each user process. Contrast with [multi-threaded server \(MTS\)](#). Net8 sends the address of an existing server process back to the client. The client will then resend its connect request to the server address provided.

default domain

The [domain](#) within which most client requests take place. It could be the domain where the client resides, or it could be a domain from which the client requests network services often. Default domain is also the client configuration parameter that determines what domain should be appended to unqualified network name requests. A name request is unqualified if it does not have a "." character within it.

delegated administration

A Net8 network where network management is delegated to one or more administrative regions below the [root administrative region](#). Also referred to as distributed or decentralized administration. Contrast with [central administration](#).

delegated administrative region

A region hierarchically below the [root administrative region](#). Any region other than the root administrative region.

destination

The client that is the endpoint of a connection. The initiator of the connection requires some data or service of the destination.

dispatcher

A *dispatcher* enables many clients to connect to the same server without the need for a dedicated server for each client. A dispatcher handles and directs multiple incoming network session requests to the [shared server](#).

distributed administration

See *delegated administration*.

distributed processing

Division of front-end and back-end processing to different computers. Net8 supports distributed processing by transparently connecting applications to remote databases.

DNS

Domain Name Services. A machine used to translate the host name specified in the *TNSNAMES.ORA file* into the host internet address (IP).

domain

A grouping of network objects, such as databases, that simplifies the naming of network services. Within a domain, all the names must be unique.

domestic domains

The set of domains that are managed within a given administrative region. Domains are only domestic relative to a region; they are never domestic in any absolute sense. Also referred to as local domains.

E**error message**

A message from a computer program informing you of a potential problem or condition preventing program or command execution.

external naming

A *net service name* resolution that uses a supported third-party naming service, such as NIS or NDS.

external procedure

Function or procedure written in a third-generation language (3GL) that can be called from PL/SQL code. Only C is supported for external procedures.

F

failover

See [connect-time failover](#).

firewall support

See [Net8 access control](#).

flat naming model

An Oracle Names infrastructure in which there is only one domain. All names must be unique within that domain.

foreign domains

The set of domains not managed within a given administrative region. Domains are only foreign relative to a region; they are not foreign in any absolute sense. A network administrator typically defines foreign domains relative to a particular region to optimize Names Server caching performance.

G

General Inter-Orb Protocol (GIOP)

GIOP is a [presentation layer](#) type that is used in Java option connections to provide messaging.

global database link

A database link that links each database in a network to all other databases. This enables any user of any database in the network to specify a global object name in a SQL statement or object definition. (The global object name for the Debunk must be the same as the database service name.)

See also [database link](#), [private database link](#), and [public database link](#).

global database name

The global database name is the full name of the database which uniquely identifies it from any other database. The global database name is of the form "[database_name.database_domain](#)". For example:

SALES.US.ORACLE.COM

The database name portion, SALES, is a simple name you wish to call your database. The database domain portion, .US.ORACLE.COM, specifies the database domain in which the database is located, making the global database name unique.

When possible, Oracle Corporation recommends that your database domain mirror the network domain.

The global database name is the default *service name* of database, as specified by the SERVICE_NAMES parameter in the *initialization file (INITSID.ORA)* file.

H

heterogeneous services

An integrated component that provides the generic technology for accessing non-Oracle systems from the Oracle server. Heterogeneous Services enables you to:

- Use Oracle SQL to transparently access data stored in non-Oracle systems as if the data resides within an Oracle server.
- Use Oracle procedure calls to transparently access non-Oracle systems, services, or application programming interfaces (APIs), from your Oracle distributed environment.

hierarchical naming model

An Oracle Names infrastructure in which names are divided into multiple hierarchically-related domains. You can use the hierarchical naming model with either central or delegated administration.

host naming

A *net service name* resolution that enables users in a TCP/IP environment to resolve net service names via their existing name resolution service. This name resolution service might be Domain Name Services (DNS), NIS or simply a centrally-maintained set of /etc/hosts files. Host Naming allows users to connect to an Oracle server by simply providing the server computer's host name or host name alias. No client configuration is required to take advantage of this feature. This method is recommended for simple TCP/IP environments.

I

instance registration

See [database instance registration](#)

initialization file (INITSID.ORA)

File that contains information to initialize the database and instance.

Inter-Orb Protocol (IIOP)

An implementation of [General Inter-Orb Protocol \(GIOP\)](#) over TCP/IP or TCP/IP with SSL for connections to the Java option.

IIOP Clients

Clients that use the [General Inter-Orb Protocol \(GIOP\)](#) presentation to access the Java option. These clients include:

- Enterprise JavaBeans (EJBs)
- CORBA Servers
- Java Stored Procedures

J

Java DataBase Connectivity (JDBC) Drivers

JDBC Drivers provide Java programmers access to an Oracle database.

OCI JDBC Driver

Oracle's Level 2 JDBC driver is targeted to client-server Java application programmers and Java-based middle-tier developers. The JDBC/OCI driver converts JDBC invocations to calls to the Oracle Call Interface (OCI) which are then sent over Net8 to the Oracle database server.

Thin JDBC Driver

Thin JDBC driver is Oracle's Type 4 driver designed for Java applet and Java application developers. The JDBC driver establishes a direct connection to the Oracle database server over Java sockets. Access to the database is assisted with a lightweight implementation of Net8 and [Two-Task Common \(TTC\)](#).

K

keyword-value pair

The combination of a keyword and a value, used as the standard unit of information in connect descriptors and many configuration files. Keyword-value pairs may be nested; that is, a keyword may have another keyword-value pair as its value.

L

listener

A separate process that resides on the server whose responsibility is to listen for incoming client connection requests and manage the traffic to the server.

The listener brokers the client request, handing off the request to the server. Every time a client (or server acting as a client) requests a network session with a server, a listener receives the actual request. If the client's information matches the listener's information, the listener grants a connection to the server.

LISTENER.ORA file

A configuration file that describes one or more listeners on a server. For a pre-release 8.1 database, the LISTENER.ORA.ORA file can also define the *Oracle System Identifier (SID)* of the database served by the listener. Release 8.1 databases do not require identification of the database because of *database instance registration*.

Listener Control Utility (LSNRCTL)

A utility included with Net8 to control various functions, such as to start, stop, and get the status of the listener.

load balancing

A feature by which client connections are distributed evenly among multiple listeners, dispatchers, instances, and nodes so that no single component is overloaded.

Load balancing happens at the following levels:

- *client load balancing*
- *connection load balancing*

local naming

A *net service name* resolution that locates network addresses by using information configured and stored on each individual client *TNSNAMES.ORA file*. Local naming is most appropriate for simple distributed networks with a small number of services that change infrequently.

location transparency

A distributed database characteristic that allows applications to access data tables without knowing where they reside. All data tables appear to be in a single database, and the system determines the actual data location based on the table name. The user can reference data on multiple nodes in a single statement, and the system automatically and transparently routes (parts of) SQL statements to remote nodes for execution if needed. The data can move among nodes with no impact on the user or application.

logging

A feature in which errors, service activity, and statistics are written to a log file. The log file provides additional information for an administrator when the error message on the screen is inadequate to understand the failure. The log file, by way of the error stack, shows the state of the software at various layers.

See also *tracing*.

M**multi-threaded server (MTS)**

A server that is configured to allow many user processes to share very few server processes so the number of users that can be supported is increased. Without MTS, each user process requires its own *dedicated server*. A new server process is created for each client requesting a connection. With MTS (also called *shared server*) configuration, many user processes connect to a *dispatcher*.

multiplexing

Combining multiple sessions for transmission over a single transport connection in order to conserve the operating system's resources. See also *connection concentration*

multi-protocol support

A feature of Oracle Connection Manager that enables a client and server with different networking protocols to communicate with each other. This feature replaces functionality previously provided by the Oracle Multi-Protocol Interchange with SQL*Net version 2.

N**naming methods**

The method used by a client application to resolve a net service name to a network address when attempting to connect to a database service. Net8 provides four naming methods:

- *host naming*
- *local naming*
- *Oracle Names*
- *external naming*

naming model

The set and structure of domains within which names can be allocated.

In a flat naming model, there is a single domain.

In a hierarchical naming model, the highest level is the root domain, and all other domains are hierarchically related.

NDS

NetWare Directory Service

net service name

A name for a connect descriptor that is easy to use and remember. End users need only know the appropriate service name to make a connection. Each connect descriptor is assigned a service name in the network definition. It identifies a:

- network route to the service, including the location of the listener through a protocol address
- *service name*(s) used to identify the release 8.1 services, or *Oracle System Identifier (SID)* used to identify a pre-release 8.1 databases
- optional instance name of the release 8.1 service to which to connect
- Before a service, such as a database, can receive connections from clients, clients must be configured to use a net service name, which

Net8

Oracle's remote data access software that enables both client-server and server-server communications across any network. Net8 supports distributed processing and distributed database capability. Net8 runs over and interconnects many communications protocols. Net8 is backward compatible with SQL*Net version 2.

Net8 Assistant

A Java-based tool that helps you to configure and administer:

Profile — The tool enables the user to create or edit the local configuration file (SQLNET.ORA) on the local machine.

Net Service Names— The tool enables the user to create or edit the local naming file (TNSNAMES.ORA).

Listeners — The tool enables the user to create or edit the local configuration file (LISTENER.ORA) on the local machine.

Oracle Names Servers — The tool enables the user to configure and control an Oracle Names server from the local machine, and to control other Oracle Names servers on remote machines.

Net8 Configuration Assistant

A Java-based tool that configures certain client and server Net8 components after an install.

Net8 Open

The application program interface (API) to Net8 that enables programmers to develop both database and non-database applications that make use of the Net8 network already deployed in their environment. Net8 Open provides applications a single common interface to all industry standard network protocols.

NetWare

A network operating system produced by Novell.

network

A group of two or more computers linked together through hardware and software to allow the sharing of data and/or peripherals.

Net8 access control

A feature of Oracle Connection Manager that sets rules for denying or allowing certain clients to access designated servers. Also known as firewall support.

network administrator

The person who performs network management tasks such as installing, configuring, and testing network components. The administrator typically maintains the configuration files, connect descriptors and service names, aliases, and public and global database links.

network character set

As defined by Oracle, the set of characters acceptable for use as values in keyword-value pairs (that is, in connect descriptors and configuration files). The set includes alphanumeric upper- and lowercase, and some special characters.

network listener

See [listener](#).

network object

Any service that can be directly addressed on a network; for example, a listener or a Oracle Names server.

network protocol

See [Oracle protocol](#).

network service

In an Oracle application network, a service performs tasks for its service consumers; for example, a Names Server provides name resolution services for clients.

network service name

A network descriptions of a service name. The network description specifies one or more network routes to the service, including any number of Net8 components such as listeners. Net service names also specify whether network connections should be load balanced among multiple listeners and/or fail over to an alternate listener if a network connection cannot be established.

Network Session (NS)

NS is a *session layer* types that is used in typical Net8 connection to establish and maintaining the connection between a client application and server.

NI

Network Interface

NIS**NL**

Network Library

NN

Network Naming (Oracle Names)

node

A computer or terminal that is part of a network.

NPI

Network Program Interface

NR

Network Routing

NS

Network Session

NT

Network Transport

O

OPI

Open Program Interface

ORACLE_HOME

An alternate name for the top directory in the Oracle directory hierarchy on some directory-based operating systems.

Oracle Connection Manager

Oracle Connection Manager is a software component that provides *connection concentration*, *multi-protocol support* and *Net8 access control*.

Oracle Names

A directory service made up of a system of Oracle Names servers that provide name-to-address resolution for each Net8 service on the network.

Oracle Names Server

A server that uses Oracle Names to store a service's network address along with its simple name so that client applications can request connections with simple names, rather than lengthy addresses.

Oracle Names infrastructure

A set of initial decisions and policies that govern how names are allocated, and how Oracle Names operates. The infrastructure defines how users and administrators interact with the Oracle Names system.

Oracle protocol

A set of rules that defines how data is transported across networks. There are several industry standard transport protocols, such as TCP/IP and SPX.

Oracle Rdb

A database for Digital's 64-bit platforms. With Net8, Oracle Rdb servers appears the same way to clients as Oracle databases. Because Oracle Rdb has its own listener, the client interacts with Rdb in the same manner as it does with Oracle7.

Oracle System Identifier (SID)

A name that identifies a specific instance of the running Oracle8i software. For any database, there is at least one instance referencing the database.

For pre-release 8.1, SID was used to identify the database. The SID is included in the CONNECT DATA part of the connect descriptor in a *TNSNAMES.ORA file*, and in the definition of the network listener in the *LISTENER.ORA file*. If you choose the default configuration, the SID defaults to "*".

P

packet

A block of information sent over the network each time a connection or data transfer is requested. The information contained in packets depends on the type of packet: connect, accept, redirect, data, etc. Packet information can be useful in troubleshooting.

parameter

Information passed to a program, command, or function, such as a file specification, a keyword, or a constant value.

password

A string (word or phrase) used for data security and known only to its owner. Passwords are entered in conjunction with an operating system login ID, Oracle username, or account name, in order to connect to an operating system or software application (such as the Oracle database). Whereas the username or ID is public, the secret password ensures that only the owner of the username can use that name, or access that data.

preferred Oracle Names Server

The Oracle Names Server(s) preferred by a client for names resolution; usually the Oracle Names Server that is physically closest to the client, or available over the least expensive network link.

presentation layer

The presentation layer manages the representation of information that application layer entities either communicate or reference in their communication. Example of session layers are *Two-Task Common (TTC)* and *General Inter-Orb Protocol (GIOP)*.

prestarted or prespawnd dedicated server process

Prespawnd dedicated server processes are prestarted by the Net8 listener before any incoming connection request. They improve the time it takes to establish a connection on servers where the multi-threaded server is not used or not supported on a given machine. They also use allocated memory and system resources better by recycling server processes for use by other connections with shutting down and recreating a server.

private database link

A database link created by one user for his or her exclusive use.

See also [database link](#), [global database link](#), and [public database link](#).

protocol stack

Designates a particular [presentation layer](#) and [session layer](#) combination.

public database link

A database link created by a DBA on a local database which is accessible to all users on that database.

See also [database link](#), [global database link](#), and [private database link](#).

R

RDBMS

Relational Database Management System

region

See [administrative region](#).

region database

Tables in an Oracle database that store Oracle Names information.

root administrative region

The highest level administrative region in a distributed installation. The root administrative region contains the root domain.

root domain

The highest level domain in a hierarchical naming model.

RPC

Remote Procedure Call

S

service replication

A process that fully replicates a directory system on the network. New services need to register with only one Names Server. The service replication process automatically distributes the new registration to all other active Names Servers on the network.

service name

A logical representation of a database, which is the way a database is presented to clients. A database can be presented as multiple services and a service can be implemented as multiple database instances. The service name is a string that is the *global database name*, a name comprised of the database name and domain name, entered during installation or database creation. If you are not sure what the global database name is, you can obtain it from the combined values of the SERVICE_NAMES parameter in the database initialization file, *INITSID.ORA*.

The service name is included in the CONNECT DATA part of the connect descriptor in a *TNSNAMES.ORA file*.

session layer

The session layer provides the services needed by the *presentation layer* entities that enable them to organize and synchronize their dialogue and manage their data exchange. This layer establishes, manages, and terminates network sessions between the client and server. An example of a session layer is *Network Session (NS)*.

shared server

See *multi-threaded server (MTS)*.

SID

See *Oracle System Identifier (SID)*.

SID_LIST_listener_name

A section of the *LISTENER.ORA file* that defines the *Oracle System Identifier (SID)* of the database served by the listener. This section is only valid for version 7.x and version 8.0 Oracle databases, as release 8.1 instance information is now automatically registered with the listener. Static configuration is also required for other services, such as external procedures and heterogeneous services, and some management tools, including Oracle Enterprise Manager version 2.

SPX

Sequenced Packet Exchange, a network protocol known for high performance and acceptance among many major network management systems, in particular, Novell Advanced NetWare.

SQL*Net

Net8's precursor. An Oracle product that works with the Oracle Server and enables two or more computers that run the Oracle RDBMS or Oracle tools such as SQL*Forms to exchange data through a network. SQL*Net supports distributed processing and distributed database capability. SQL*Net runs over and interconnects many communications protocols.

system or topology data

Data used by the Oracle Names server to control regular functioning or communicate with other Oracle Names servers. Includes interchanges, root region's Oracle Names servers, and any delegated regions' Oracle Names servers.

T**TNS**

See *Transparent Network Substrate (TNS)*.

TNSNAMES.ORA file

A file that contains connect descriptors mapped to service names. The file may be maintained centrally or locally, for use by all or individual clients.

tracing

A facility that writes detailed information about an operation to an output file. The trace facility produces a detailed sequence of statements that describe the events of an operation as they are executed. Administrators use the trace facility for diagnosing an abnormal condition; it is not normally turned on.

See also *logging*.

Transparent Network Substrate (TNS)

A foundation technology, built into Net8, Oracle Connection Manager and Oracle Names, that works with any standard network transport protocol.

Two-Task Common (TTC)

TTC is a *presentation layer* type that is used in typical Net8 connection to provide character set and data type conversion between different character sets or formats on the client and server.

U**username**

The name by which a user is known to the Oracle Server and to other users. Every username is associated with a password, and both must be entered to connect to an Oracle database.

UPI

User Program Interface

W**well-known Oracle Names server**

Addresses for one or more Oracle Names servers are hardcoded into both the Oracle Names server and its clients. Oracle Names servers then become available at these well known addresses, so that clients do not need to be told, by way of configuration files, where to find the server.

Symbols

- " symbol
 - reserved, in configuration files, C-4
- # symbol
 - reserved, in configuration files, C-4
- (, 4-11, C-65
- () symbol
 - reserved, in configuration files, C-4
-), 9-7
- = symbol
 - reserved, in configuration files, C-4
- | (slash) symbol
 - reserved, in configuration files, C-4
- ' symbol
 - reserved, in configuration files, C-4

Numerics

- 1521 port, 4-23, 6-16, 6-27, 6-66, 6-80
- 1521 port, if not using for default listener port, 4-23, 6-67
- 1575 port, 2-45, 6-32, 6-80, C-65
 - Oracle Names server, 2-45
- 1630 port, 2-46, 4-14, 6-80, 7-31, 7-35, 7-38, B-9, C-76
- 1646 port, C-26
- 1830, 4-14
- 1830 port, 4-14, 4-19, 5-21, 6-80, 7-31, 7-35
- 2481 port, 6-80, 7-58, 7-59
- 2481 port, if not using for default, 4-23, 7-59
- 2482 port, 6-80, 7-58, 7-59
- 2482, if not using for default, 4-23, 7-59
- 2483 port, 6-80

A

- ACT parameter, 7-39, C-80
- ACTIVE_DELAYS, A-91
- ADDRESS (ADD or ADDR) attribute, 7-63, 9-4
- ADDRESS parameter, 6-12, 6-76, 7-55, C-5, C-39, C-52
- ADDRESS_LIST parameter, 4-11, 6-77, C-6, C-39, C-53
- addresses, configuring, 6-64
- administrative regions
 - defined, 6-23
 - delegated, 2-40
 - delegated, below root, 2-42
 - delegating domains, 6-50
 - described, 2-33
 - in Oracle Names, 2-33
 - root, 2-40
- Advanced Program-to-Program Communication (APPC), 4-26
- Always Use Dedicated Server option, 7-26
- ANSWER_TIMEOUT parameter, C-78
- APPC architecture, 4-26
- application program interface (API), for non-Oracle data sources, 9-2
- applications
 - building with Net8 OPEN, 9-8
 - configuring system to use custom, 9-9
 - samples provided with Net8 OPEN, 9-11
- architecture
 - Net8, 2-5
 - Oracle Connection Manager, 2-46
 - Oracle Names, 2-31
- ARGS parameter, 6-78

- ARGV0 parameter, 6-78
- A.SMD record, 6-68, 8-17, A-45
- asynchronous data operations, 2-4
- Attempts Per Names Server option, 6-39
- audit trail
 - described, 11-25
 - script for using information, 11-26
- authentication, using Oracle Advanced Security, 7-28
- AUTHENTICATION_LEVEL parameter, C-78
- AUTOMATIC_IPC parameter, 4-20, 5-19, 7-26

B

- backslash, defined, xx
- BACKUP parameter, C-44
- benefits
 - provided by TNS, 2-6
- bequeath, 9-15
 - connecting with, 8-8
 - described, 4-27
 - event programming, 9-14
 - parameters for addresses, 6-78
- BEQUEATH_DETACH parameter, 7-24, 9-15, C-7
- bequeathed sessions, 2-25
- biometrics
 - see Oracle Advanced Security
- buffer flushing, described, 3-14
- building custom applications with Net8 OPEN, 9-8

C

- CDS value, 7-19
- Cell Directory Service (CDS), 3-6
- centralized naming
 - advantages and disadvantages, 3-9
 - described, 3-5
 - establishing a connection with, 3-5
 - recommended for, 3-9
- CHANGE_PASSWORD command, A-4
- character set
 - for net service name, C-5
 - network, for keyword values, C-4
- child process termination, 9-15
- Choose Start >, defined, xx

- CKCFG.ORA file, 6-25
- CKPCCH.ORA file, 6-25
- CKPREG.ORA file, 6-25
- client
 - configuring a profile, 7-15
 - configuring domains, 7-16, 7-21
 - configuring for connection concentration, 7-33
 - configuring for multi-protocol support, 7-36
 - configuring for Net8 access control, 7-40
 - configuring for Oracle Connection Manager
 - address, 7-43
 - configuring for Oracle Connection Manager features, 7-41
 - configuring for Oracle Rdb database, 7-71
 - configuring with local naming, 6-20
 - configuring with the Net8 Configuration Assistant after install, 4-24
 - discovering Oracle Names servers, 6-37
 - load balancing, 2-30, 4-10, 7-11, C-41
 - log files, 11-20
 - methods of connecting, 8-8
 - migrating, 5-19
 - overriding multi-threaded server mode, 9-11
 - randomization, 2-30, C-41
 - session with multiple protocols, diagram, 2-50
 - SQLNET.LOG, 11-20
 - testing connectivity, 8-20
 - testing using 3GL, 8-9
 - testing using application, 8-9
 - testing using OS, 8-9
 - testing using special commands, 8-10
 - trace files, 11-33
 - upgrading, 5-19
- client application layer, described, 2-10
- client load balancing, 2-30, 4-10, C-41
 - configuring, 7-11
 - using to improve network performance, 3-12
- Client Registration ID option, 7-23
- client troubleshooting, 11-4
- CMADM_PID.LOG file contents, 11-18, 11-27
- CMADM_PID.TRC file contents, 11-32
- CMADMIN process, 4-14, 5-21
 - purpose, 2-47
- CMAN parameter, C-76
- CMAN_ADMIN parameter, 4-14, 4-19, 5-21

CMAN_PID.LOG file contents, 11-18, 11-27
 CMAN_PID.TRC file contents, 11-32
 CMAN_PROFILE parameter, C-77
 CMAN_RULES parameter, 7-39, C-80
 CMAN.LOG file, 11-18
 CMAN.LOG file contents
 , 11-18
 CMAN.ORA
 see Oracle Connection Manager, configuration file
 CMCTL
 purpose, 2-47
 see Oracle Connection Manager Control Utility
 CMGW process
 purpose, 2-46
 CNAME.SMD record, 8-17, A-45
 coexistence
 Oracle Names, 5-10
 Oracle8i Client r. 8.0/SQL*Net Client connection to Oracle8i database r. 8.1, 5-6
 Oracle8i Client r. 8.1 connecting to Oracle8i r. 8.0 or Oracle7 database, 5-8
 Oracle8i Client r. 8.1 connection to Oracle8i database r. 8.1, 5-5
 SERVICE_NAME parameter in TNSNAMES.ORA, 5-7
 SID parameter in TNSNAMES.ORA, 5-7
 third-party Oracle7 applications, 5-7
 coexistence issues, 5-5 to 5-10
 command line
 CMCTL commands from, A-89
 LSNRCTL commands from, A-3
 NAMESCTL commands from, A-31
 commands
 CMCTL reference, A-90 to A-93
 LSNRCTL reference, A-4 to A-30
 NAMESCTL reference, A-34 to A-88
 comments in configuration files, C-3
 communications, stack, 2-6
 COMMUNITY parameter, 5-4
 compatibility
 Oracle DCE Integration with Net8, 7-29
 TRCROUTE utility with earlier versions of SQL*Net, 8-23
 configuration file
 listener (LISTENER.ORA), B-6
 local naming (TNSNAMES.ORA), B-5
 Oracle Connection Manager (CMAN.ORA), B-9
 Oracle Names servers (NAMES.ORA), B-7
 profiles (SQLNET.ORA), B-2
 protocol addresses, 6-76
 syntax rules, C-2
 configuring
 a profile, 7-15
 client for connection concentration, 7-33
 client for multi-protocol support, 7-36
 client for Net8 access control, 7-40
 client load balancing, 7-11
 client Oracle Connection Manager, 7-41
 clients and servers to use Oracle Names, 6-37
 clients for local naming, 6-20
 CMAN.ORA for connection concentration, 7-31
 CMAN.ORA for multi-protocol support, 7-35
 CMAN.ORA for Net8 access control, 7-38
 CONNECT_DATA attributes, 7-7
 connection concentration, 7-31
 connection pooling, 7-49
 connect-time failover, 7-11
 CORBA Servers, 7-51 to 7-64
 domains, 7-16, 7-21
 Enterprise JavaBeans (EJBs), 7-51 to 7-64
 external naming, 3-7, 6-5
 external procedures, 7-65
 heterogeneous services, 6-71
 host naming, 6-5, 6-7, 6-51
 IIOP clients, 7-50, 7-51 to 7-64
 Inter-Orb Protocol (IIOP) clients, 7-50
 Java option, 7-50
 Java Stored Procedures, 7-51
 listener names, 6-69
 listeners, 6-62 to 6-75
 listeners for local naming, 6-21, 6-40
 local naming, 6-5, 6-12
 multiple addresses, 7-3
 multiple domains, 6-29, 6-34
 multi-protocol support, 7-35
 net service names for external naming, 6-51
 net service names for local naming, 6-14
 Net8 access control, 7-38

- Net8 OPEN, 9-8
- Oracle Advanced Security, 7-29
- Oracle Names, 6-5, 6-21 to 6-50
- Oracle Names servers, 6-27
- overview, 6-2
- prespawnd dedicated servers, 6-73
- protocol addresses, 6-64, 6-76
- region database for Oracle Names, 6-26, 6-31
- replicating data among Oracle Names servers, 6-32
- routing connection requests in a profile, 7-26
- server for connection concentration, 7-32
- servers for local naming, 6-14
- system to use custom applications, 9-9
- the listener with Oracle Database Configuration Assistant, 4-23
- to connect to Heterogeneous Services, 7-67
- to connect to Oracle Rdb database, 7-71
- with Oracle Names, 6-4
- with the Net8 Configuration Assistant, 4-22, 4-24
- without Oracle Names, 6-4
- connect descriptors, 2-2, 6-12, C-31
- connect operations, 2-2
- connect strings, 6-2, 8-2
- CONNECT_DATA parameter, 6-12, 7-7, C-43
- CONNECT_TIMEOUT parameter, 11-13, C-62
- connecting
 - bequeath, 8-8
 - common problems, 8-11
 - connect strings, 6-2, 8-2
 - entering a connect string, 8-8
 - host naming, 8-8
 - local naming, 8-8
 - methods of connecting, 8-8
 - Oracle Names, 8-8
 - Oracle7 connections to Oracle8 databases, 5-6
 - Oracle8i Client r. 8.0/SQL*Net Client to Oracle8i database r. 8.1, 5-6
 - Oracle8i Client r. 8.1 to Oracle8i database r. 8.1, 5-5
 - Oracle8i Client r. 8.1 to Oracle8i r. 8.0 or Oracle7 database, 5-8
 - proper startup of components, 8-2
 - release 8.0/7.x service, 5-14
 - release 8.1 service, 5-12
 - routing, 1-6
 - with dedicated server, 7-26
 - with Oracle Connection Manager, 7-26
- connection concentration, 2-48, 4-30, 7-31
 - advantages relative to connection pooling, 3-12
 - using to improve network performance, 3-11
- connection load balancing, 2-30, 4-2
- connection pooling
 - advantages relative to connection concentration, 3-12
 - using to improve network performance, 2-50, 3-11
 - using with multi-threaded servers, 7-49
- connections, 7-52
 - adjusting listener queue size to avoid errors, 3-14, 6-70, 8-13
 - concurrent, increasing number of, 6-70
 - connect descriptors, 2-2
 - connect strings, 1-3
 - dedicated server, described, 2-23
 - listener, 2-28
 - multi-threaded server, described, 2-20
 - requests, extending size of backlog, 8-13
 - sessions-based IOP, 7-52
 - to services, 1-3
 - verifying, 8-25
 - with net service names, 1-3
 - with Oracle Connection Manager, 1-6
- CONNECTIONS (CON or CONN) attribute, 7-49, 9-5
- connect-time failover, 2-29, C-40
- connect-time failover, configuring, 7-11
- control utilities
 - described, 8-15, A-1
 - Listener Control Utility (LSNRCTL), A-3
 - Oracle Connection Manager Control Utility (CMCTL), A-89
 - Oracle Names Control Utility (NAMESCTL), A-31
- CORBA Servers
 - configuring, 7-51 to 7-64
 - described, 7-50

- creating
 - Oracle Names server with region database, 6-27
 - Oracle Names server without region database, 6-32
- Custom install type, 4-21
- CyberSAFE authentication method
 - see Oracle Advanced Security

D

- DAEMON.TRACE_DIRECTORY parameter, C-7
- DAEMON.TRACE_LEVEL parameter, C-7
- DAEMON.TRACE_MASK parameter, C-8
- Data Encryption Standard
 - see Oracle Advanced Security
- data operations
 - asynchronous, 2-4
 - synchronous, 2-4
- data transfer, maximizing, 3-13
- Database, 2-29
- database
 - as a repository for Oracle Names, 2-34
 - distributed, 2-5
- database instance registration
 - benefits, 2-27
 - client load balancing, 2-30
 - connection load balancing, 2-30
 - connect-time failover, 2-29
 - described, 2-27, 4-2
 - load balancing, 2-30
 - MTS dispatcher registration, 2-27
 - service registration, 2-27, 6-2
- database links, 2-36, 6-22, 6-44
 - global database link, 2-36, 6-22
 - private database, 2-36, 6-22
 - public database, 2-36, 6-22
- DB_DOMAIN parameter, 4-5, 5-17
- DB_NAME parameter, 5-17
- DBAccess Interface, 2-18
- DBSNMP_START command, A-5
- DBSNMP_STATUS command, A-5
- DBSNMP_STOP command, A-6
- DCE Integration, described, 4-32
- DDO
 - see Dynamic Discovery Option

- dead connection detection
 - described, 2-3
 - limitations, 7-23
 - SQLNET.EXPIRE_TIME, C-20
- dedicated servers
 - bequeathed sessions, 2-25
 - described, 2-23
 - multi-threaded servers contrasted with, 9-2
 - prespawned dedicated servers, 2-23
 - routing connections, 7-26
 - with multi-threaded servers, 9-10
- default domain
 - in Oracle Names, 2-38
 - null, 7-16
- Default Domain option, 6-39
- DELEGATE_DOMAIN command, 6-50, A-34
- delegated administrative regions
 - below root, 2-42
 - diagram, 2-41
 - in Oracle Names, 2-40
 - domains to, 6-50
- DESCRIPTION parameter, 6-12, C-38, C-51
- DESCRIPTION_LIST parameter, C-38, C-52
- diagnosing
 - client, 11-4
 - errors, see troubleshooting
 - server, 11-3
- Disable Out-of-Band Break option, 7-24
- DISABLE_OOB parameter, 7-24
- disabling UNIX signal handler, 9-15
- disconnecting
 - abnormal termination, 2-3
 - additional connect request, 2-3
 - from servers, 2-3
 - user initiated, 2-3
- discovering Oracle Names servers, 2-45, 6-36
- dispatchers
 - connections for IIOP clients, 7-53, 7-63
 - described, 2-20
 - number to start, 9-8
 - setting the number of, 9-8
- DISPATCHERS (DIS or DISP) attribute, 7-49, 7-63, 9-5

- Distributed Computing Environment (DCE)
 - Integration
 - described, 7-29
- distributed databases, 2-5
- distributed processing, 2-5
- DLCR.RDBMS.OMD record, 8-17, A-45
- DL.RDBMS.OMD record, 8-17, A-45
- DNS, described, Glossary-7
- domain, 4-5
- Domain Name Services, described, Glossary-7
- DOMAIN_HINT command, A-35
- domains
 - default, 2-38
 - delegating, 6-50
 - described, 2-33, 6-23
 - multiple hierarchically related, 2-39, 6-29, 6-34
 - naming considerations, 2-33
 - required by root administrative regions, 2-40
 - with a single domain naming structure,
 - diagram, 2-37
- DST parameter, 7-39, C-80
- DUMP_TNSNAMES command, 5-24, A-36
- Dynamic Discovery Option, 2-43
 - migrating issues, 5-24

E

- Enterprise JavaBeans (EJBs)
 - configuring, 7-51 to 7-64
 - described, 7-50
- error messages
 - 12203, 11-13
 - 12535, 11-13
 - 12547, 11-13
 - 20002-20021 for Net8 OPEN, 9-12
 - contacting Oracle Customer Support, 11-58
 - example of trace data, 11-42
 - in trace file, 11-42
 - NNL-00018, 6-31
 - NNL-00024, 6-31
 - ORA-00508, 11-9
 - ORA-1017, 11-3
 - ORA-1034, 8-13, 11-3
 - ORA-12154, 8-13, 11-8
 - ORA-12198, 11-8
 - ORA-12203, 8-12, 11-8, 11-9
 - ORA-12203, sample error stack, 11-17
 - ORA-12203, troubleshooting, 8-12
 - ORA-12224, 11-11
 - ORA-12500, 11-11
 - ORA-12533, 11-11
 - ORA-12538, 11-9
 - ORA-12541, 8-13
 - ORA-12545, 11-12
 - ORA-12560, 11-12
 - ORA-3113, 11-12
 - ORA-3121, 11-12
 - resolving, 11-1 to 11-58
 - TNS-0116, 8-14
 - TNS-01169, 8-14
 - using log file to track, 11-25
- error stack
 - described, 11-16
 - entries in log files, 11-25
 - sample, 11-17
 - typical layers, 11-17
- establishing a session
 - through multi-threaded servers, 2-20
 - using bequeath, 8-8
 - using centralized naming, 3-5
 - using external naming, 3-6
 - using host naming, 3-4, 8-8
 - using local naming, 3-5, 8-8
 - using Oracle Names, 3-5, 8-8
- exception operations, described, 2-4
- EXIT command
 - of CMCTL, reference, A-90
 - of LSNRCTL, reference, A-6
 - of NAMESCTL, reference, A-37
- external, 6-71

external naming

- advantages and disadvantages, 3-9
- Cell Directory Service, 3-6
- configuring, 3-7, 6-51
- described, 1-4, 3-6, 6-5
- establishing a connection with, 3-6
- features, 4-31
- recommended for, 3-9
- using NetWare Directory Service (NDS), 4-31
- using Network Information Service, 3-6, 6-51
- using Network Information Service (NIS), 4-31
- using with Oracle Names, 3-7

external procedures

- configuring, 7-65
- described, 7-65

F

failover, C-43

- connect-time, 2-29, 4-12, 7-11, C-40
- transparent application failover, 2-29, C-43
- with Oracle Enterprise Manager, 2-29, 6-71, 7-12

FAILOVER parameter, 2-29, 4-12, 4-15, 5-11, 7-11, 7-12, 7-14, C-40

FAILOVER_MODE parameter, C-43

finger utility, 9-11

firewall support, 2-49, 4-30

- Net8 Firewall Proxy, 2-49
- Oracle Connection Manager, 2-49

FLUSH command, A-38

FLUSH_NAME command, A-39

forward slash, defined, xx

ftp sample in Net8 OPEN, 9-11

G

General Inter-Orb Protocol (GIOP)

see GIOP

GIOP, C-55

- described, 2-14
- dispatcher configuration, 7-63
- listener configuration, 7-56
- oracle.aurora.server.GiopServer, 7-52
- oracle.aurora.server.SGiopServer, 7-52
- presentation, 7-51

presentation layer, 4-13, C-51, C-55

global database links, 2-36, 6-22, 6-44

- adding link qualifiers, 6-44, 6-46
- registering with Oracle Names, 6-45

global database name

- configuring on the listener, 6-71, C-58
- defined, 6-3

described, 1-3, 4-9, 5-17, 6-71, C-58

Global Database Name field, 7-71

Global Database Name option, 6-71, 7-8

GLOBAL_DBNAME parameter, 6-71, C-58

GLOBAL_NAME parameter, 7-8, 7-71, C-46

H

Handler Name option, 7-7

HANDLER_NAME parameter, 7-7

HELP command

- of LSNRCTL, reference, A-7
- of NAMESCTL, reference, A-40

Heterogeneous Services

- configuring to connect, 7-67
- described, 7-67

hierarchical naming model

- described, 2-38
- diagram, 2-38

- host naming
 - advantages and disadvantages, 3-8
 - configuring, 6-7
 - connecting with, 8-8
 - described, 1-4, 6-5
 - establishing a connection with, 3-4, 8-8
 - recommended for, 3-8
 - requirements, 6-7
- HOST parameter, 6-79
- HOSTNAME value, 7-19
- HOSTS file, 2-43
- HS parameter, 7-7, C-46

I

- Identix authentication method
 - see Oracle Advanced Security
- IIOp, 4-13
 - configuring for SSL, 7-64
 - configuring the MTS_DISPATCHER parameter, 7-54
 - described, 7-50, 7-51
 - GIOP presentation layer, 2-14, 4-13
 - no session layer, 2-14
- IIOp clients
 - 2481 port for listener, 4-23, 7-59
 - 2482 port for listener, 4-23, 7-59
 - configuring, 7-50, 7-51 to 7-64
 - configuring a non-default listener port, 4-23, 7-59
 - configuring for SSL, 7-64
 - connections directly to dispatchers, 7-53, 7-63
 - connections through listeners, 7-52
 - CORBA Servers, 7-50
 - Enterprise Java Beans (EJBs), 7-50
 - GIOP presentation layer, 2-14, 4-13, 9-7, C-55
 - Java Stored Procedures, 7-50
 - no session layer, 2-14
 - oracle.aurora.server.GiopServer presentation layer, 7-52
 - oracle.aurora.server.SGiopServer presentation layer, 7-52
 - regular IIOp connections, 7-52
 - session-based connections, 7-52
 - SSL support
 - SSL support for IIOp clients, 7-53
 - IIOp clients connections through listeners, 7-54
 - IIOp stack communications
 - described
 - Implementing, 6-71, 7-12
 - improving network performance
 - by adjusting SDU size, 3-13
 - by client load balancing, 3-12
 - by connection pooling, 2-50, 3-11
 - by randomizing client requests, 2-30, C-41
 - by using connection concentration, 3-11
 - with listener queuesize, 3-14
- INITDB_NAME.ORA file
 - LOCAL_LISTENER parameter, 6-67
- Initial Preallocated Requests option, 6-39
- INITSID.ORA file
 - DB_DOMAIN parameter, 5-17
 - DB_NAME parameter, 5-17
 - INSTANCE_NAME parameter, 5-17, 6-4
 - LOCAL_LISTENER parameter, 4-23, 6-16, 6-66, 6-67
 - MTS_DISPATCHERS parameter, 9-4
 - SERVICE_NAMES parameter, 5-17, 6-4
- installation
 - Custom, 4-21
 - default configuration, 4-22
 - default configuration for local naming, 6-13
 - Minimal, 4-21
 - Net8 Client, 4-24
 - Net8 Server, 4-22
 - Oracle Names, 6-24
 - overview, 4-21
 - Typical, 4-21
- instance name
 - described, 6-3
 - verifying, 5-17
- Instance Name option, 7-7
- INSTANCE_NAME parameter, 4-3, 4-9, 4-16, 5-17, 6-3, 6-4, 7-7, C-47
- INTCHG.ORA file, migration issues, 5-22
- Internet, 2-14
- Internet Inter-Orb Protocol (IIOp)
 - see IIOp

Inter-Orb Protocol (IIOP)
 see IIOP
IPC, parameters for addresses, 6-78
italic, defined, xx

J

Java DataBase Connectivity (JDBC) Drivers
 see JDBC
Java DBAccess Interface layer, described, 2-18
Java NS, described, 2-18
Java option
 configuring, 7-50
 connections, 7-50
 dispatcher configuration, 7-63
 listener configuration, 7-54, 7-56
Java Sockets, described, 2-18
Java Stored Procedures
 configuring, 7-51
 described, 7-50
Java VM, described, 7-51
JavaNet, described, 2-18
JavaTTC, described, 2-18
JDBC
 drivers, 2-16
 Java DBAccess Interface, 2-18
 Java NS, 2-18
 Java Sockets, 2-18
 JavaNet, 2-18
 JavaTTC, 2-18
 OCI, 2-16
 TCP/IP Protocol, 2-18
 Thin, 2-16

K

Kerberos authentication method
 see Oracle Advanced Security
KEY parameter, 6-78
keyword syntax rules, for configuration files, C-3
keyword values, network character set for, C-4

L

link layer, described, 2-8
LISTENER, 1-3, 4-22, 6-6
listener, 6-71
 adding an address, 7-3
 adjusting queue size for, 3-14, 6-70, 8-13
 Audit Trail and log files for, 11-25
 client connections, described, 2-28
 configuration file, 1-3
 new parameters in release 8.1, 4-18
 configuration file changes in release 8.1, 4-13
 configuration file sample, B-6
 configuration file, described, 6-6
 configuration for Oracle Names, 6-40
 configuration parameter reference, C-50 to C-64
 configuration with local naming, 6-21
 configuring, 6-62 to 6-75
 configuring a name, 6-69
 configuring a non-default port for IIOP
 clients, 4-23, 7-59
 configuring a non-default port for Net8
 clients, 4-23, 6-67
 configuring address for Oracle Connection
 Manager, 7-46
 configuring for external procedures, 6-71, 7-65,
 7-66
 configuring for host naming, 6-8
 configuring for Oracle Enterprise
 Manager, 2-29, 4-13, 4-23, 5-20, 6-71, 7-12,
 C-40
 configuring global database name, 6-71, C-58
 configuring listener addresses, 6-76
 configuring Oracle Home Directory, 6-72
 configuring Oracle System Identifier (SID), 6-71
 configuring protocol addresses, 6-64
 configuring prespawed dedicated servers, 6-73
 configuring with the Net8 Configuration
 Assistant, 4-22
 configuring with the Oracle Database
 Configuration Assistant, 4-23
 connections for IIOP clients, 7-52, 7-54
 connect-time failover, 2-29, 4-12
 control utility, see Listener Control Utility
 (LSNRCTL)

- default configuration, 6-62
- deleting protocol address, 7-5
- handling concurrent connections, 6-70
- in a typical Net8 connection, diagram, 2-28
- increasing queue size, 3-14, 6-70
- instance registration, 2-27
- load balancing, 2-30
- log files, 11-15, 11-21
- partial address listen, 2-23
- service registration, 2-27
- starting, 8-5
- stopping, error TNS-01169, 8-14
- testing, 8-18
- trace files, 11-36
 - transparent application failover, 2-29
- LISTENER (LIS or LIST) attribute, 7-60, 9-5
- listener addresses, 6-76
- LISTENER attribute, 4-23, 6-66, 7-59, 7-60
- Listener Control Utility (LSNRCTL)
 - command reference, A-4 to A-30
 - described, 8-15
 - function of and syntax format, A-3
 - SET PASSWORD command, 8-14
 - starting a listener, 6-64, 8-5
 - stopping a listener, 6-64
 - testing the network
- listener load balancing, using TRCROUTE, 8-24
- LISTENER.LOG file contents, 11-18
- LISTENER.ORA
 - see listener, configuration file
- LISTENER.TRC file contents, 11-32
- LLU parameter, 6-78
- LLU_NAME parameter, 6-78
- Load, 5-25
- load balancing
 - client load balancing, 2-30, C-41
 - connection load balancing, 2-30
- LOAD_BALANCE parameter, 2-30, 4-10, 4-17, 5-11, 7-11, C-41
- LOAD_TNSNAMES command, 5-25, 5-28, 6-43, A-41
- loading TNSNAMES.ORA into Oracle Names, 5-28, 6-43
- local naming
 - advantages and disadvantages, 3-8
 - client configuration for, 6-20
 - client load balancing, 4-10
 - configuration file
 - new parameters in release 8.1, 4-15 to 4-17
 - configuration file changes in release 8.1, 4-9 to 4-12
 - configuration file sample, B-5
 - configuration file, described, 6-6
 - configuration parameter reference, C-31 to C-49
 - configuring, 6-12
 - configuring protocol addresses, 6-64
 - configuring with Net8 Assistant, 6-14
 - configuring with Net8 Configuration Assistant, 6-13
 - connecting with, 8-8
 - connect-time failover, 4-12
 - default configuration, 6-13
 - described, 1-4, 6-5
 - establishing a connection with, 3-5, 8-8
 - listener configuration for, 6-21
 - recommended for, 3-8
 - server configuration for, 6-14
- LOCAL_LISTENER parameter, 4-23, 6-16, 6-66, 6-67, 7-59, 7-60, 9-5
- LOCAL_LU parameter, 6-78
- LOCAL_LU_NAME parameter, 6-78
- LOG, 11-20, 11-22
- log files
 - CMADM_PID.LOG, 11-22
 - CMAN_PID.LOG, 11-22
 - CMAN_PID.TRC, 11-37
 - default names for, 11-18
 - for listener, 11-25
 - LISTENER.LOG, 11-21
 - NAMES.LOG, 11-21
 - SQLNET.LOG for clients and servers, 11-20
 - using to track errors, 11-25
- LOG_DIRECTORY parameter, 11-21, C-62
- LOG_DIRECTORY_CLIENT parameter, 11-20, C-8
- LOG_DIRECTORY_SERVER parameter, 11-20, C-8
- LOG_FILE parameter, 11-21, C-62
- LOG_FILE_CLIENT parameter, 11-20, C-9
- LOG_FILE_SERVER parameter, 11-20, C-9

LOG_LEVEL parameter, 11-22, C-78
 LOG_STATS command, A-41, A-42
 loopback test, 11-4
 lowercase courier, defined, xx
 LSNRCTL
 see Listener Control Utility (LSNRCTL)
 LU_NAME parameter, 6-78
 LU6.2
 described, 4-26
 parameters for addresses, 6-78
 supporting networks, 4-26
 supporting topologies, 4-26

M

MAX_FREELIST_BUFFERS parameter, C-79
 maximizing data transfer, by adjusting SDU size, 3-13
 Maximum Open Connections option, 6-39
 Maximum Prespawned Servers option, 6-73
 Maximum Wait Each Attempt option, 6-39
 MAXIMUM_CONNECT_DATA parameter, C-79
 MAXIMUM_RELAYS parameter, C-79
 MD5, 4-31
 MDN parameter, 6-78
 METHOD parameter, C-44
 migrations, 5-1 to 5-30
 defined, 5-16
 INTCHG.ORA file, 5-22
 LISTENER.ORA changes, 5-20
 modifying configuration files, 5-19
 NAMESINI.SQL script for Oracle Names, 5-29
 NAMESUPG.SQL script, 5-27
 NAMESUPG.SQL script for Oracle Names, 5-23, 5-29
 of software, 5-18
 of TNSNAMES.ORA with Net8 Assistant, 5-11
 Oracle Connection Manager, 5-21
 Oracle Names, 5-23 to 5-30
 SQLNET.ORA changes, 5-19
 TNSNAMES.ORA changes, 5-19
 TNSNAV.ORA file, 5-22
 TNSNET.ORA file, 5-22
 to Oracle Names from ROSFILES, 5-26
 to Oracle Names v. 8 from Oracle Names with Dynamic Discovery Option, 5-24
 to Oracle Names, using a region database, 5-23
 verifying service name and instance name, 5-17
 Minimal install type, 4-21
 MODE parameter, 6-78
 MOST_RELAYS, A-91
 MTS, 9-4
 MTS_DISPATCHERS parameter, 7-32, 7-49, 9-4
 ADDRESS attribute, 7-63, 9-4
 CONNECTIONS attribute, 7-49, 9-5
 DISPATCHERS attribute, 7-49, 7-63, 9-5
 LISTENER, 7-60
 LISTENER attribute, 4-23, 6-66, 7-59, 7-60, 9-5
 MULTIPLEX attribute, 7-32, 9-6
 POOL attribute, 7-49, 9-6
 PRESENTATION attribute, 7-52, 7-54, 7-57, 7-63, 9-7
 PROTOCOL attribute, 7-32, 7-49, 7-54, 7-57, 9-4
 SERVICE attribute, 9-7
 SESSIONS attribute, 9-7
 setting initially, 9-8
 TICKS attribute, 7-49, 9-7
 MTS_SERVICE parameter, 9-7
 multiple addresses, 7-3, 7-11
 configuring client load balancing, 7-12
 configuring connect-time failover, 7-12
 MULTIPLEX (MUL or MULT) attribute, 7-32, 9-6
 multiplexing, 4-30
 used in connection concentration, 3-11
 Multi-Protocol Interchange, 5-2
 multi-protocol support, 2-49, 4-30, 5-2, 5-22
 configuring, 7-35

- multi-threaded servers
 - configuring dispatchers, 9-8
 - connection load balancing, 2-30
 - connection pooling, 2-50
 - dedicated server contrasted with, 9-2
 - described, 2-4, 2-20
 - dispatchers, 2-20
 - initialization parameters, 9-4
 - routing session requests to, 2-20
 - starting, 9-4
 - using with connection pooling, 7-49
 - using with Oracle Connection Manager, 2-48, 7-31
 - with dedicated servers, 9-10

N

Named Pipes

- described, 4-26
- parameters for addresses, 6-78
- NAMES_DIRECTORY_PATH parameter, 6-9, 6-19, 6-54, 6-57, 6-60
- NAMES.ADDRESSES parameter, 6-28, 6-33, C-65
- NAMES.ADMIN_REGION parameter, 5-25, 6-28, C-66
- NAMES.AUTHORITY_REQUIRED parameter, C-66
- NAMES.AUTO_REFRESH_EXPIRE parameter, C-67
- NAMES.AUTO_REFRESH_RETRY parameter, C-67
- NAMES.CACHE_CHECKPOINT_FILE parameter, C-68
- NAMES.CACHE_CHECKPOINT_INTERVAL parameter, C-68
- NAMES.CONFIG_CHECKPOINT_FILE parameter, C-68
- NAMES.CONNECT_TIMEOUT parameter, C-67
- NAMESCTL
 - see Oracle Names Control Utility
- NAMESCTL.INTERNAL_ENCRYPT_PASSWORD parameter, C-13
- NAMESCTL.INTERNAL_USE parameter, C-13
- NAMESCTL.NO_INITIAL_SERVER parameter, C-13

- NAMESCTL.NOCONFIRM parameter, C-13
- NAMESCTL.SERVER_PASSWORD parameter, C-14
- NAMESCTL.TRACE_DIRECTORY parameter, C-14
- NAMESCTL.TRACE_FILE parameter, C-14
- NAMESCTL.TRACE_LEVEL parameter, C-14
- NAMESCTL.TRACE_UNIQUE parameter, C-15
- NAMES.DCE.PREFIX parameter, C-9, C-12
- NAMES.DEFAULT_DOMAIN parameter, 6-15, 6-39, 7-16
- NAMES.DEFAULT_FORWARDERS parameter, C-69
- NAMES.DEFAULT_FORWARDERS_ONLY parameter, C-69
- NAMES.DEFAULT_ZONE parameter, 5-4
- NAMES.DEFAULT.DOMAIN parameter, C-9
- NAMES.DIRECTORY_PATH parameter, C-10
 - CDS, 7-19
 - HOSTNAME, 7-19
 - NDS, 7-19
 - NIS, 7-19
 - ONAMES, 7-19
 - TNSNAMES, 7-19
- NAMES.DOMAIN_CHECKPOINT_FILE parameter, 4-20
- NAMES.DOMAIN_HINTS parameter, C-69
- NAMES.DOMAINS parameter, 6-29, 6-30, 6-34, C-70
- NAMES.FORWARDING_AVAILABLE parameter, C-70
- NAMES.FORWARDING_DESIRED parameter, C-70
- NAMESINI, 6-26
- NAMESINI.SQL script, 5-24, 6-26
- NAMES.INITIAL_RETRY_TIMEOUT parameter, 6-39
- NAMES.INITIAL_RETRY_TIMEOUT parameter, C-10
- NAMES.KEEP_DB_OPEN parameter, C-70
- NAMES.LOG file contents, 11-18
- NAMES.LOG_DIRECTORY parameter, 11-21, C-71
- NAMES.LOG_FILE parameter, 11-21, C-71

NAMES.LOG_STATS_INTERVAL
 parameter, C-71
 NAMES.LOG_UNIQUE parameter, C-71
 NAMES.MAX_OPEN_CONNECTIONS
 parameter, 6-39
 NAMES.MAX_OPEN_CONNECTIONS
 parameter, C-72
 NAMES.MAX_REFORWARDS parameter, C-72
 NAMES.MESSAGE_POOL_START_SIZE
 parameter, 6-39
 NAMES.MESSAGE_POOL_START_SIZE
 parameter, C-11, C-72
 NAMES.NO_MODIFY_RESPONSE
 parameter, C-73
 NAMES.NO_REGION_DATABASE
 parameter, C-73
 NAMES.ORA
 see Oracle Names servers, configuration file
 NAMES.PASSWORDS parameter, C-73
 NAMES.PREFERRED_SERVERS parameter, 7-21,
 C-12
 NAMES.REGION_CHECKPOINT_FILE
 parameter, C-73
 NAMES.REQUEST_RETRIES parameter, 6-39,
 C-12
 NAMES.RESET_STATS_INTERVAL
 parameter, C-74
 NAMES.SAVE_CONFIG_ON_STOP
 parameter, C-74
 NAMES.SERVER_NAME parameter, 6-28, 6-33
 NAMES.SERVER_NAMES parameter, C-74
 NAMES.TOPOLOGY_CHECKPOINT_FILE
 parameter, 4-20
 NAMES.TRACE_DIRECTORY parameter, 11-37,
 C-74
 NAMES.TRACE_FILE parameter, 11-37, C-75
 NAMES.TRACE_FUNC parameter, C-75
 NAMES.TRACE_LEVEL parameter, C-75
 NAMES.TRACE_UNIQUE parameter, 11-37, C-75
 NAMES.TRC file contents, 11-32
 NAMESUPG.SQL script, 5-23, 6-26
 NAMESUPG.SQL script for Oracle Names, 5-27
 NAMES.USE_PLUG_AND_PLAY parameter, 4-20

naming methods
 centralized naming, 3-5
 choosing, 3-8
 configuring with the Net8 Configuration
 Assistant, 4-23
 described, 1-4, 3-3
 external naming, 1-4, 3-6, 4-31, 6-5, 6-51, 7-19
 host naming, 1-4, 3-3, 6-5, 6-7, 7-19
 local naming, 1-4, 3-5, 6-5, 6-12, 7-19
 Oracle Names, 1-4, 3-5, 6-5, 6-21, 7-19
 prioritizing order, 7-18
 naming model
 hierarchical, 2-38
 single domain, 2-37
 NDS
 see NetWare Directory Service
 nested addresses, 4-11
 net service names
 adding an address, 7-3
 addresses, resolving, 3-3
 configure an Oracle Connection Manager
 address, 7-43
 configuring CONNECT_DATA, 7-7
 configuring protocol addresses, 6-64
 defined, 1-3, 2-2
 deleting, 7-6
 deleting protocol address, 7-5
 described, 4-4, 6-2
 external naming, 1-4, 3-6
 host naming, 3-3
 identifying a service, 6-3
 local naming, 1-4, 3-5
 modifying, 7-2
 multiple addresses, 7-3, 7-11
 Oracle Names, 1-4, 3-5
 prioritizing naming methods, 7-18
 registering with Oracle Names, 6-42
 renaming, 7-3
 testing with TNSPING, 8-22

- Net8, 2-12, 6-13, 7-58
 - compatibility with Oracle DCE Integration, 7-29
 - concept described, 1-2
 - connect operations, 2-2
 - control utilities, 8-15
 - data operations, 2-4
 - differences from SQL*Net, 5-2
 - exception operations, 2-4
 - in stack communications, 2-12
 - installing, 4-21
 - JDBC drivers, 2-16
 - operations, 2-4
 - overview, 1-2
 - primary functions, 2-4
- Net8 access control, 2-49, 4-30, 7-38
- Net8 Assistant, 5-28, 6-43, 7-71
 - adding addresses, 7-3
 - Always Use Dedicated Server option, 7-26
 - Attempts Per Names Server option, 6-39
 - Client Registration ID option, 7-23
 - configure static listener information, 6-71
 - configuring a listener name, 6-69
 - configuring a non-default listener in Oracle Names, 6-68, 7-62
 - configuring advanced profile options, 7-23
 - configuring an Oracle Rdb Database, 7-72
 - configuring client for Oracle Connection Manager, 7-43, 7-46
 - configuring client load balancing, 7-11
 - configuring client profiles, 7-15
 - configuring CONNECT_DATA, 7-8
 - configuring domains, 7-16
 - configuring external procedures, 7-65
 - configuring failover, 7-11
 - configuring for a release 8.0 service, 5-14
 - configuring for a release 8.1 service, 5-12
 - configuring Heterogeneous Services, 7-67
 - configuring local naming method, 6-14
 - configuring log files, 11-23
 - configuring multiple domains for an Oracle Names server, 6-29, 6-34
 - configuring net service names, 6-14
 - configuring Oracle Advanced Security, 7-28, 7-29
 - configuring Oracle Names in a region database environment, 6-27
 - configuring Oracle8 database for Heterogeneous Services, 7-67, 7-72
 - configuring preferred Oracle Names servers
 - Oracle Names servers
 - preferred, 7-20
 - configuring prespawnded dedicated servers, 6-74
 - configuring protocol addresses, 6-64
 - configuring the listener for IIOP clients, 7-58
 - configuring the listener with service information, 6-72
 - configuring trace files, 11-38
 - Default Domain option, 6-39
 - delegating domains to administrative regions, 6-50
 - deleting protocol addresses, 7-5
 - described, 4-29, 5-2
 - Disable Out-of-Band Break option, 7-24
 - discovering Oracle Names servers, 6-36, 6-37
 - Global Database Name option, 6-71, 7-8, 7-71
 - Handler Name option, 7-7
 - Initial Preallocated Requests option, 6-39
 - Instance Name option, 7-7
 - loading TNSNAMES.ORA data into Oracle Names server, 5-25
 - loading TNSNAMES.ORA file into Oracle Names, 5-25
 - Maximum Open Connections option, 6-39
 - Maximum Prespawnded Servers option, 6-73
 - Maximum Wait Each Attempt option, 6-39
 - migrating TNSNAMES.ORA file, 5-11
 - migrating TNSNAMES.ORA files, 5-11
 - multiple address options, 7-11
 - Oracle Home option, 7-69
 - Oracle RDB Database option, 7-8
 - Prefer Connection Manager Routing option, 7-26
 - prioritizing naming methods, 7-18
 - Program Name option, 7-69
 - RDB Database option, 7-71
 - Register Service with Oracle Names option, 6-40
 - registering aliases with Oracle Names, 6-49

- registering global database links with Oracle Names, 6-45
- registering net service names with Oracle Names, 6-42
- routing connection requests, 7-26
- Session Data Unit (SDU) option, 7-7
- setting advanced connection options, 7-8
- SID option, 7-69
- starting, 4-29
- starting an Oracle Names server, 6-31, 6-36, 8-3
- testing network objects, 8-17
- Timeout option, 6-74
- TNS Time Out Value option, 7-23
- Turn Off UNIX Signal Handling option, 7-24
- Type of Service option, 7-8
- upgrading TNSNAMES.ORA file, 5-11
- Use Dedicated Server option, 7-7
- Use for Heterogeneous Services option, 7-7
- Use Options Compatible with Net8 8.0 Clients option, 5-11
- Use Oracle8i Release 8.0 Compatible Identification option, 5-11
- verifying connectivity, 8-25

Net8 Client

- described, 4-24, 4-25
- installing, 4-24

Net8 Configuration Assistant, 4-28

- configuring local naming method, 6-18
- configuring naming methods, 4-23
- configuring net service names, 6-18
- configuring the listener, 4-22
- configuring the local naming method, 6-13
- deleting net service names, 7-6
- described, 4-5, 4-28, 5-2
- modifying net service name attributes, 7-2
- renaming net service names, 7-3
- starting, 4-28
- using after installation, 4-22, 4-24

Net8 Easy Config

- described, 4-28, 5-2

Net8 Firewall Proxy, 2-49

Net8 OPEN

- API function calls, 9-3
- compatibility with C language, 9-3
- configuration requirements, 9-8
- configuring system to use applications, 9-9
- error messages, 9-12
- finding API, 9-8
- for distributed applications, 9-2
- integrating with non-SQL information, 9-2
- library, 9-8
- sample applications, 9-11

Net8 Server

- described, 4-22, 4-25
- installing, 4-22

Net8 stack communications

- client application, 2-10
- Oracle protocols, 2-12
- session, 2-12
- TTC, 2-11, 9-7
- Two-Task Common (TTC) presentation, 2-11

NetWare Directory Service (NDS), 3-6, 6-51, 7-19

NetWare Directory Service (NDS) External Naming

- described, 4-31

Network Authentication (NA)

- layer in error stacks, 11-17
- layer in stack communications, 2-12

network character set, keyword values, C-4

network configuration

- centralized management, 6-4
- localized management, 6-4

Network Encryption (NAE), layer in error stacks, 11-17

Network Information Service (NIS), 3-6, 6-51, 7-19

Network Information Service (NIS) External Naming, 4-31

Network Interface (NI)

- described, 2-12
- layer in error stacks, 11-17

Network Naming (NN)

- layer in error stacks, 11-17
- layer in stack communications, 2-12

Network Program Interface (NPI), 2-13

Network Routing (NR)

- layer in error stacks, 11-17
- layer in stack communications, 2-12

- Network Security
 - DES_40, 4-31
 - DES_56, 4-31
 - described, 4-31
 - MD5, 4-31
 - RC4_128, 4-31
 - RC4_40, 4-31
 - RC4_56, 4-31
- Network Services (NA), layer in error stacks, 11-17
- Network Session, 4-13
- Network Session (NS), 4-13, C-51
- Network Session (NS), layer in error stacks, 11-17
- Network Transport (NT), layer in error stacks, 11-17
- NI
 - see Network Interface (NI)
- NIS
 - see Network Information Service
- NIS Maps, 6-52
- NMCBUILD.SQL script, 5-27
- NMCCRNT.SQL script, 5-27
- NNL-00018 error message, 6-31
- NNL-00024 error message, 6-31
- no session layer, 4-13, 7-55
- nodes, 2-6
- NPI
 - see Network Program Interface
- NS session, 4-13, C-51
- NS.SMD record, 8-17, A-45

O

- obsolete parameters in release 8.1, 4-20, 5-4
- ONAMES value, 7-19
- Open System Interconnection (OSI) model
 - client application layer, described, 2-7
 - described, 2-6
 - link layer, 2-8
 - physical, 2-8
 - presentation layer, 2-7
 - session layer, 2-8
 - transport layer, 2-8

- operations
 - connect, 2-2
 - data, 2-4
 - exception, 2-4
 - Net8, 2-4
- OPI
 - see Oracle Program Interface
- ORA-00508 error message, 11-9
- ORA-1017 error messages, 11-3
- ORA-1034 error message, 8-13
- ORA-1034 error messages, 11-3
- ORA-12154 error message, 8-13, 11-8
- ORA-12198 error message, 11-8
- ORA-12203 error message, 8-12, 11-8, 11-9, 11-13
 - sample error stack, 11-17
 - troubleshooting, 8-12
- ORA-12224 error message, 11-11
- ORA-12500 error message, 11-11
- ORA-12533 error message, 11-11
- ORA-12535 error message, 11-13
- ORA-12538 error message, 11-9
- ORA-12541 error message, 8-13
- ORA-12545 error message, 11-12
- ORA-12547 error message, 11-13
- ORA-12560 error message, 11-12
- ORA-3113 error message, 11-12
- ORA-3121 error message, 11-12
- Oracle Advanced Security
 - authentication, 7-28
 - biometrics authentication method, 7-28
 - configuring, 7-28, 7-29
 - CyberSAFE authentication method, 4-32
 - DCE Integration, 4-32, 7-29
 - DES_40, 4-31
 - DES_56, 4-31
 - described, 4-31
 - encryption, 7-28
 - features of, 4-31
 - Identix authentication method, 4-32
 - Kerberos authentication method, 4-32, 7-28
 - MD5, 4-31
 - Network Security, 4-31
 - RADIUS authentication method, 4-32, 7-28
 - RC4_128, 4-31
 - RC4_40, 4-31

- RC4_56, 4-31
- Secure Sockets Layer (SSL) authentication
 - method, 7-28
- SecurID authentication method, 4-32
- security features, 7-28
- Single Sign-On, 4-32
- SSL authentication method, 4-32
- using Oracle Advanced Security, 7-28
- using with Oracle Connection Manager, 7-28
- with Data Encryption Standard, 7-28
- with RSA Data Security RC4, 7-28
- Oracle Call Interface (OCI) layer, described, 2-11
- Oracle Connection Manager
 - CMADMIN process, 2-47
 - CMCTL process, 2-47
 - CMGW process, 2-46
 - configuration file
 - new parameters in release 8.1, 4-19
 - configuration file changes in release 8.1, 4-14
 - configuration file sample, B-9
 - configuration parameter reference, C-76 to C-80
 - configuring address, 6-76
 - configuring address for listener, 7-46
 - configuring address for Oracle Connection Manager, 7-43
 - configuring CMAN.ORA for connection concentration, 7-31
 - configuring CMAN.ORA for multi-protocol support, 7-35
 - configuring CMAN.ORA for Net8 access control, 7-38
 - configuring connection concentration, 7-31
 - configuring multi-protocol support, 7-35
 - configuring Net8 access control, 7-38
 - connection concentration, 2-48, 3-11, 4-30
 - described, 4-30
 - log files, 11-22
 - log files, understanding, 11-27
 - migrating, 5-21
 - migration issues, 5-21
 - multiple protocol support, 2-49
 - multi-protocol support, 4-30, 5-22
 - Net8 access control, 4-30
 - Net8 access control, 2-49
 - Net8 Firewall Proxy, 2-49
 - processes, 2-46
 - role in multiple protocol environments,
 - diagram, 2-50
 - routing connections, 1-6, 7-26
 - SOURCE_ROUTE parameter, C-41
 - starting, 8-7
 - testing, 8-18
 - TNSNAMES.ORA file, 7-34
 - trace files, 11-37
 - upgrading, 5-21
 - using instead of Oracle Multi-Protocol Interchange, 5-22
 - using with multi-threaded servers, 2-48, 7-31
- Oracle Connection Manager Control Utility (CMCTL)
 - command reference, A-90 to A-93
 - function and syntax format, A-89
 - starting, 8-7
 - testing the network
 - using, 8-16
- Oracle Customer Support, contacting, 11-58
- Oracle Database Configuration Assistant
 - configuring listener for Oracle Enterprise Manager, 4-23
- Oracle Database Configuration Assistant,
 - configuring the listener, 4-23
- Oracle DCE Integration
 - compatibility with Net8, 7-29
 - described, 7-29
- Oracle Enterprise Manager
 - configuring static information in the listener, 4-23
 - connect-time failover, 2-29, 6-71, 7-12
 - SID_LIST parameter in LISTENER.ORA, 2-29, 4-13, 5-20, 6-71, 7-12, C-40
 - static service information in LISTENER.ORA, 2-29, 5-20, 6-71, 7-12, C-40
- Oracle Home Directory
 - configuring on the listener, 6-72
 - described, 6-72
- Oracle Home Directory option
 - Net8 Assistant
 - Oracle Home Directory, 6-72
 - Oracle home directory, defined, xx
 - Oracle Home option, 7-69

Oracle Names

- administrative regions, 2-33
- advantages and disadvantages, 3-9
- architecture, 2-31
- centralized naming, 3-5
- client cache, 8-4
- coexistence, 5-10
- configure clients and servers, 6-37
- configuring, 6-21 to 6-50
- configuring a region database, 6-26, 6-31
- configuring multiple domains, 6-29, 6-34
- connecting with, 8-8
- creating Oracle Names servers, 6-25
- default domain, 2-38
- delegated administrative regions, 2-42
- delegating domains, 6-50
- described, 2-31, 4-30, 6-5
- differences between versions, 2-42
- discovering Oracle Names servers, 6-36, 6-37
- discovery, 2-45
- domains, 2-33
- Dynamic Discovery Option, 2-43
- establishing a connection with, 3-5
- hierarchical naming model, 2-38
- installing, 6-24
- listener configuration for, 6-40
- loading TNSNAMES.ORA file into Oracle Names, 5-25
- log files, 11-21
- migrating, 5-23 to 5-30
- migration, 5-10
- multiple domains, 2-39
- NAMESINI.SQL script, 5-24, 5-29, 6-26
- NAMESUPG.SQL script, 5-23, 5-27, 5-29, 6-26
- naming network components, 2-37
- querying objects, 8-17
- recommended for, 3-9
- region database, 2-34
- regions, 2-33
- registering global database link qualifiers, 6-44, 6-46
- registering global database links, 6-44
- registering net service names, 6-41
- registering objects, 6-41
- replicating data, 2-34, 6-32
- root administrative region, 2-40
- .SDNS.ORA file, 2-45
- SDNS.ORA file, 2-45
- single domain naming mode, 2-37
- starting from the command line, 8-17
- starting from the Control Panel, 8-7
- storing aliases, 2-36, 6-22
- storing database global database names and addresses, 2-35, 6-21
- storing database links, 2-36, 6-22
- storing information in a region database, 6-26
- storing net service names, 2-35, 6-22
- storing Oracle Connection Manager information, 2-36, 6-22
- storing other Oracle Names server names and addresses, 2-35, 6-21
- trace files, 11-37
- upgrading, 5-23 to 5-30
- using, 6-21
- using a database as a repository for information, 2-34
- using multiple regions, 2-40
- using with external naming, 3-7
- version 1, 2-42
- version 2, 2-43
- version 8, 2-44

Oracle Names Control Utility (NAMESCTL)

- command reference, A-34 to A-88
- confirmation mode, A-33
- delegating domains, 6-50
- description of and types of commands, A-31
- discovering Oracle Names servers, 6-36, 6-37
- distributed operation, A-32
- loading TNSNAMES.ORA into Oracle Names, 5-25, 5-28, 6-43
- modes of operation, A-2
- parameter options, A-31
- querying objects, 8-17
- registering aliases, 6-49
- registering global database links, 6-45
- registering net service names, 6-42
- security, A-3, A-32
- SET and SHOW modifiers, A-3, A-31
- starting an Oracle Names server, 6-31, 6-36, 8-3
- testing network objects with, 8-17
- testing the network, 8-15

Oracle Names servers

- CKCFG.ORA file, 6-25
- CKPCCH.ORA file, 6-25
- CKPREG.ORA file, 6-25
- configuration file sample, B-7
- configuration file, described, 6-6
- configuration parameter reference, C-65 to C-75
- configuring multiple domains, 6-29, 6-34
- configuring protocols addresses, 6-64, 6-76
- creating, 6-25
- creating for a region database, 6-27
- creating for replication, 6-32
- data it can store, 2-35
- data stored, 2-35
- delegating domains, 6-50
- described, 1-5, 2-31
- discovery of Oracle Names servers, 6-36
- in delegated administrative regions, 2-40
- in root administrative region, 2-40
- loading TNSNAMES.ORA file into Oracle Names, 5-25
- log files, 11-21
- preferred, 2-42

registering

- global database link qualifiers, 6-46
- registering global database link qualifiers, 6-44
- registering global database links, 6-44
- registering net service names, 6-41
- registering objects, 6-41
- replicating data, 6-32
- starting, 6-31, 8-3
- starting client cache, 8-4
- storing aliases, 2-36, 6-22
- storing database global database names and addresses, 2-35, 6-21
- storing database links, 2-36, 6-22
- storing net service names, 2-35, 6-22
- storing Oracle Connection Manager information, 2-36, 6-22
- storing other Oracle Names server names and addresses, 2-35, 6-21
- testing, 8-16
- testing with TNSPING, 8-22
- trace files, 11-37
- well-known, 2-43

Oracle Network Manager, described, 5-2

Oracle Parallel Server

- client load balancing, 4-10
- configuring client load balancing, 7-12
- configuring connect-time failover
- connect-time failover, 2-29, 4-15
- FAILOVER parameter, 4-15, C-40
- FAILOVER_MODE parameter, C-43
- INSTANCE_NAME parameter, 4-16, C-47
- LOAD_BALANCE parameter, 4-17, C-41

Oracle Program Interface (OPI), 2-13

Oracle Program Interface (OPI), described, 2-13

Oracle protocols

- Bequeath, 4-27
- configuring addresses, 6-78
- described, 2-12, 4-25
- LU6.2, 4-26
- Named Pipes, 4-26
- SPX, 4-25
- TCP/IP, 4-25
- TCP/IP with SSL, 4-6, 4-25

- Oracle Rdb database
 - configuring for connection, 7-71
 - described, 7-71
- Oracle RDB Database option, 7-8
- Oracle System Identifier (SID), configuring on the listener, 6-71
- ORACLE_HOME parameter, 6-72, 7-66, 7-69, C-58
- ORACLE_HOME, defined, xx
- oracle.aurora.server.GiopServer, 9-7
- oracle.aurora.server.SGiopServer, 7-52
- OSS.SOURCE.LOCATION parameter, C-15
- OSS.SOURCE.MY_WALLET parameter, 7-64, C-15
- OUT_OF_RELAY, A-91

P

packet

- example of trace data, 11-41
- types of, 11-41

parameters

- listener configuration reference, C-50 to C-64
- local naming configuration
 - reference, C-31 to C-49
- new in CMAN.ORA file for release 8.1, 4-19
- new in LISTENER.ORA file for release 8.1, 4-18
- new in TNSNAMES.ORA file for release 8.1, 4-15 to 4-17
- obsolete for release 8.1, 4-20
- obsolete with Net8, 5-4

- Oracle Connection Manager configuration
 - reference, C-76 to C-80

- Oracle Names servers configuration
 - reference, C-65 to C-75

- Oracle protocols, 6-78
- profile configuration reference, C-7 to C-30
- protocol configuration reference, C-81

- PARTNER_LU_LOCAL_ALIAS parameter, 6-78

- PARTNER_LU_NAME parameter, 6-78

password

- for NAMESCTL access, A-3, A-32
- required to stop the listener, 8-14

- PASSWORD command, A-43

- PASSWORDS parameter, C-62

- physical layer, described, 2-8

- PING utility, 8-16, A-44

- testing Oracle Names servers with, 8-16

- PIPE parameter, 6-78

planning

- overview, 3-2

- summary, 3-15

- PLU parameter, 6-78

- PLU_LA parameter, 6-78

- POOL attribute, 7-49, 9-6

- POOL_SIZE parameter, 6-73

- POOL_SIZE parameter, role in listener-created server processes, 2-23

- port number range, 6-80

- PORT parameter, 6-80

- Prefer Connection Manager Routing option, 7-26

- preferred Oracle Names servers, 2-42

- PRESENTATION (PRE or PRES) attribute, 7-52, 7-54, 7-57, 7-63, 9-7

presentation layer

- described, 2-7

- GIOP, 2-14, 4-13, 4-18, 7-51, 7-52, 9-7, C-51, C-55

- oracle.aurora.server.GiopServer, 9-7

- oracle.aurora.server.SGiopServer, 7-52, 9-7

- TTC, 2-11, 4-18, 9-7, C-51, C-55

- Two-Task Common (TTC), 2-11, 4-18, 9-7, C-51, C-55

- PRESENTATION parameter, 4-18, 7-55, C-55

- PRESPAWN_DESC parameter, C-61

- PRESPAWN_LIST parameter, C-61

- PRESPAWN_MAX parameter, 2-24, 6-73, C-60
 - role in listener-created server, 2-24

- prespawned dedicated servers, 2-23

- configuring, 6-73

- number of, 6-73

prestarted dedicated servers

- advantages and disadvantages, 3-10

- recommended for, 3-10

- private database links, 2-36, 6-22

- Procedure, 7-65

- profiles (SQLNET.ORA)
 - configuration file sample, B-2
 - configuration parameter reference, C-7 to C-30
 - configuring, 7-15
 - configuring advanced options, 7-23
 - configuring domains, 7-16
 - configuring preferred Oracle Names servers, 7-20
 - description, 6-6
 - prioritizing naming methods, 7-18
 - routing connection requests, 7-26
- Program Name option, 7-69
- PROGRAM parameter, 6-78, 7-66, 7-69, C-59
- PROTOCOL (PRO or PROT) attribute, 7-32, 7-49, 7-54, 7-57, 9-4
- PROTOCOL parameter, 6-73, 6-78
- PROTOCOL_STACK parameter, 4-18, 7-55, C-54
- protocol.EXCLUDED_NODES* parameter, C-81
- protocol.INVITED_NODES* parameter, C-81
- PROTOCOL.ORA, C-81
- protocols
 - configuration parameter reference, C-81
 - configuring address with, 6-76
 - configuring addresses, 6-78
 - selecting for network layout, 3-2
 - selecting multiple for network layout, 3-2
- protocol.VALIDNODE_CHECKING* parameter, C-81
- public database links, 2-36, 6-22

Q

- query, 6-36, 8-17
- QUERY command, 6-36, 8-17
 - A.SMD record, 6-68, 8-17, A-45
 - CNAME.SMD, 8-17, A-45
 - DLCR.TDBMS.OMD record, 8-17, A-45
 - DL.RDBMS.OMD, 8-17, A-45
 - NS.SMD, 8-17, A-45
 - of NAMESCTL, reference, A-45
 - V1ADD.NPO.OMD, 8-17, A-45
- queue size, adjusting for connection requests, 8-13
- queuesize, 3-14, 6-70
- QUEUESIZE parameter, 6-70, 8-13
 - for adjusting listener queue size, 3-14, 6-70, 8-13

- QUIT command
 - of LSNRCTL, reference, A-8
 - of NAMESCTL, reference, A-47
 - see STOP command for CMCTL
 - see STOP NOW command for CMCTL

R

- RADIUS authentication method
 - see Oracle Advanced Security
- randomizing requests among listeners, 2-30, C-41
- RAW session layer, 4-13, 7-55
- RDB Database option, 7-71
- RDB_DATABASE parameter, 7-8, 7-71, C-47
- receiving data
 - asynchronously, 2-4
 - synchronously, 2-4
- reference
 - for CMCTL commands, A-90 to A-93
 - for listener configuration, C-50 to C-64
 - for local naming configuration, C-31 to C-49
 - for LSNRCTL commands, A-4 to A-30
 - for NAMESCTL commands, A-34 to A-88
 - for Oracle Connection Manager configuration, C-76 to C-80
 - for Oracle Names servers configuration, C-65 to C-75
 - for profile configuration, C-7 to C-30
 - for protocol configuration, C-81
- region database
 - configure database server, 6-26, 6-31
 - create Oracle Names server, 6-27
 - defined, 2-34
 - storing Oracle Names information, 6-26
- regions
 - delegated administrative, 2-40
 - delegated administrative below root, 2-42
 - delegated administrative, diagram, 2-41
 - in Oracle Names, 2-33, 2-40
 - organizing multiple administrative network, 2-40
 - root administrative, 2-40

- REGISTER command, 6-42, 6-45, 6-49, A-47
- Register Service with Oracle Names option, 6-40
- registering
 - global database link qualifiers with Oracle Names, 6-44
 - global database links with Oracle Names, 6-44
 - net service names with Oracle Names, 6-41
 - objects with Oracle Names, 6-41
- regular IIOP connections, 7-52
- RELAY_STATISTICS parameter, C-79
- release 8.1 CMAN.ORA parameters, 4-19
- release 8.1 features, 4-2
 - changes in CMAN.ORA file, 4-14
 - changes in LISTENER.ORA file, 4-13
 - changes in TNSNAMES.ORA file, 4-9 to 4-12
 - connection load balancing, 4-2
 - database instance registration, 4-2
 - DB_DOMAIN parameter default, 4-5
 - Java Option support, 4-5
 - Net8 Configuration Assistant, 4-5
 - new parameters in CMAN.ORA file, 4-19
 - new parameters in LISTENER.ORA file, 4-18
 - new parameters in TNSNAMES.ORA file, 4-15 to 4-17
 - RADIUS authentication method
 - service naming, 4-3
 - SSL authentication method
 - TCP/IP with SSL protocol, 4-6
- release 8.1 LISTENER.ORA parameters, 4-18
- release 8.1 TNSNAMES.ORA parameters, 4-15 to 4-17
- RELOAD command
 - of LSNRCTL, reference, A-8
 - of NAMESCTL, reference, A-48, A-49
- REORDER_NS command, 5-25, 6-36, 6-37, 7-20
- REPEAT command, A-50
- replication of data in Oracle Names, 2-34, 6-32
- RESET_STATS command, A-51
- resolving
 - errors, see also troubleshooting
 - net service name addresses, 3-3
- RESTART command, A-52

- root
 - administrative region domain
 - requirements, 2-40
 - administrative region in Oracle Names, 2-40
 - administrative region Oracle Names servers
 - requirements, 2-40
 - regions, data definition requirements, 2-40
- root administrative regions, described, 2-40
- root domains, described, 2-40
- ROSBILD.SQL script, 5-27
- ROSFILES
 - NMCBILD.SQL script, 5-27
 - NMCGRNT.SQL script, 5-27
 - ROSBILD.SQL script, 5-27
 - ROSGRNT.SQL script, 5-27
- ROSFILES, migrating to Oracle Names, 5-26
- ROSGRNT.SQL script, 5-27
- routing connections, 1-6, 7-26
- RSA Data Security
 - see Oracle Advanced Security
- rules, syntax for configuration files, C-2

S

- SAVE_CONFIG command, A-9
- schema extension, in NDS, 6-55
- .SDNS.ORA file, 2-45
- SDNS.ORA file, 2-45
- SDU
 - configuring, 7-7
 - see session data unit
- SDU parameter, 7-7, C-48
- Secure Sockets Layer (SSL) authentication method
 - see Oracle Advanced Security
- SecurID authentication method
 - see Oracle Advanced Security
- security, NAMESCTL utility, A-3, A-32
- sending data
 - asynchronously, 2-4
 - synchronously, 2-4

- server
 - configure for external procedures, 7-66
 - configuring a profile, 7-15
 - configuring a region database for Oracle Names, 6-26
 - configuring both multi-threaded servers and dedicated servers, 9-10
 - configuring for connection concentration, 7-32
 - configuring for IIOP connections, 7-51
 - configuring for Java option, 7-51
 - configuring listeners to prespawn dedicated servers, 6-73
 - configuring with local naming, 6-14
 - configuring with the Net8 Configuration Assistant after install, 4-22
 - disconnections, 2-3
 - discovering Oracle Names servers, 6-37
 - in stack communications, 2-13
 - log files, 11-20
 - loopback test, 11-4
 - methods of connecting, 8-8
 - migrating, 5-18
 - starting the database, 8-6
 - testing, 8-18
 - to server interaction, described, 2-13
 - trace files, 11-33
 - troubleshooting, 11-3
 - upgrading, 5-18
- SERVER parameter, 6-78, 7-7, C-48
- servers
 - dedicated
 - multi-threaded contrasted with, 9-2
 - multi-threaded
 - dedicated contrasted with, 9-2
- service, 6-2, C-5
 - described, 6-3
- SERVICE (SER or SERV) attribute, 9-7
- service name
 - and address storage, in NDS, 6-56
 - character set keyword values, C-5
 - described, 2-2, 6-3
 - verifying, 5-17
- service naming, described, 4-3
- SERVICE parameter, 6-79
- SERVICE_NAME parameter, 4-3, 4-9, 4-17, 5-7, 5-19, 6-3, C-49
- SERVICE_NAMES parameter, 5-17, 6-4, 9-7
- services, 1-3
 - connecting, 1-3
 - described, 1-3
 - global database name, 1-3
 - service names, 2-2
- SERVICES command, 2-21, A-10
- session data unit
 - adjusting to improve network performance, 3-13
 - configuring, 7-7
- Session Data Unit (SDU) Size option, 7-7
- session layer
 - described, 2-8, 2-12
 - Network Naming (NN), 2-12
 - Network Routing (NR), 2-12
 - Network Session (NS), 4-13, C-51
 - NS, 2-12, 4-13, C-51
 - RAW, 4-13, 7-55
 - Transparent Network Substrate (TNS), 2-12
 - Two-Task Common (TTC), 2-12
- SESSION parameter, 4-18, 7-55, C-56
- session-based IIOP, 7-52
- SESSIONS (SES or SESS) attribute, 9-7
- SET CACHE_CHECKPOINT_INTERVAL command, A-54
- SET command, A-12, A-53
- SET CONNECT_TIMEOUT command, A-14
- SET CURRENT_LISTENER command, A-14
- SET DEFAULT_DOMAIN command, A-55
- SET DISPLAYMODE command
 - of LSNRCTL, reference, A-16
- SET FORWARDING_AVAILABLE command, A-56
- SET LOG_DIRECTORY command, A-17
- SET LOG_FILE command, A-17
- SET LOG_FILE_NAME command, A-57
- SET LOG_STATS_INTERVAL command, A-58
- SET LOG_STATUS command, A-18
- SET NAMESCTL_TRACE_LEVEL command, A-59

SET PASSWORD command
 of LSNRCTL, 8-5
 of LSNRCTL, reference, A-18
 of NAMESCTL, reference, A-60
 SET RAWMODE command
 of LSNRCTL, reference, A-19
 SET REQUESTS_ENABLED command, A-61
 SET RESET_STATS_INTERVAL command, A-62
 SET SAVE_CONFIG_INTERVAL command, A-63
 SET SAVE_CONFIG_ON_STOP command, A-20
 of NAMESCTL, reference, A-63
 SET SERVER command, A-64
 SET STARTUP_WAITTIME command, A-20
 SET TRACE_FILE_NAME command, A-65
 SET TRACE_LEVEL command, A-66
 SET TRC_DIRECTORY command, A-21
 SET TRC_FILE command, A-21
 SET TRC_LEVEL command, A-22
 shared servers
 advantages and disadvantages, 3-10
 described, 2-20
 recommended for, 3-10
 see also multi-threaded servers
 SHOW CACHE_CHECKPOINT_INTERVAL
 command, A-67
 SHOW command, A-24
 SHOW CONNECT_TIMEOUT command, A-24
 SHOW CURRENT_LISTENER command, A-24
 SHOW DEFAULT_DOMAIN command, A-69
 SHOW DISPLAYMODE command, A-24
 SHOW FORWARDING_AVAILABLE
 command, A-68
 SHOW LOG_DIRECTORY command, A-24
 SHOW LOG_FILE command, A-24
 SHOW LOG_FILE_NAME command, A-70
 SHOW LOG_STATS_INTERVAL command, A-71
 SHOW LOG_STATUS command, A-24
 SHOW NAMESCTL_TRACE_LEVEL
 command, A-72
 SHOW PASSWORD command, A-24
 SHOW RAW_MODE command, A-24
 SHOW REQUESTS_ENABLED command, A-73
 SHOW RESETS_STATS_INTERVAL
 command, A-74
 SHOW SERVER command, A-74
 SHOW STARTUP_WAITTIME command, A-24
 SHOW STATUS command, A-75
 SHOW SYSTEM_QUERIES command, A-76
 SHOW TRACE_FILE_NAME command, A-77
 SHOW TRACE_LEVEL command, A-78
 SHOW TRC_DIRECTORY command, A-24
 SHOW TRC_FILE command, A-24
 SHOW TRC_LEVEL command, A-24
 SHOW USE_PLUGANDPLAY command, A-24
 SHOW VERSION command, A-79
 SHOW_TNS_INFO parameter, C-79
 SHOWSAVE_CONFIG_ON_STOP
 command, A-24
 SHUTDOWN command, A-80
 SID field, 7-69
 SID parameter, 4-9, 5-7, C-49
 SID, configuring on the listener, 6-71
 SID_DESC parameter, C-60
 SID_LIST parameter, 4-13, 6-8, C-57, C-60
 SID_NAME parameter, 6-71, 7-66, 7-69, C-59
 signal handling
 requirements for Oracle's operating system
 dependent call, 9-14
 UNIX considerations, 9-13
 using in conjunction with Net8, 9-14
 single domain naming model, 2-37
 Single Sign-On, described, 4-32
 SNA, 4-26
 SOURCE_ROUTE parameter, 4-12, 5-11, 7-11, 7-12,
 7-34, 7-37, 7-42, C-41
 SPAWN command, A-25
 SPX
 described, 4-25
 parameters for addresses, 6-79
 SQL*Net and Net8 differences, 5-2
 SQL*Net Easy Configuration, described, 5-2
 SQLNET.ALTERNATE_PORT parameter, C-23
 SQLNET.AUTHENTICATION_KERBEROS5_
 SERVICE parameter, C-17
 SQLNET.AUTHENTICATION_SERVICES
 parameter, C-16
 SQLNET.AUTHENTICATION_TIMEOUT
 parameter, C-24
 SQLNET.CLIENT_AUTHENTICATION
 parameter, C-26

SQLNET.CLIENT_REGISTRATION
 parameter, 7-23, C-17

SQLNET.CRYPTO_CHECKSUM_CLIENT
 parameter, C-17

SQLNET.CRYPTO_CHECKSUM_SERVER
 parameter, C-18

SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT
 parameter, C-18

SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER
 parameter, C-18

SQLNET.CRYPTO_SEED parameter, C-19

SQLNET.ENCRYPTION_CLIENT parameter, C-19

SQLNET.ENCRYPTION_SERVER parameter, C-19

SQLNET.ENCRYPTION_TYPES_CLIENT
 parameter, C-20

SQLNET.ENCRYPTION_TYPES_SERVER
 parameter, C-20

SQLNET.EXPIRE_TIME parameter, 7-23, C-20

SQLNET.IDENTIX_FINGERPRINT_DATABASE
 parameter, C-21

SQLNET.IDENTIX_FINGERPRINT_DATABASE_
 PASSWORD parameter, C-21

SQLNET.IDENTIX_FINGERPRINT_DATABASE_
 USER parameter, C-21

SQLNET.IDENTIX_FINGERPRINT_METHOD
 parameter, C-21

SQLNET.KERBEROS5_CC_NAME
 parameter, C-22

SQLNET.KERBEROS5_CLOCKSKEW
 parameter, C-22

SQLNET.KERBEROS5_CONF parameter, C-22

SQLNET.KERBEROS5_KEYTAB parameter, C-23

SQLNET.KERBEROS5_REALMS parameter, C-23,
 C-25

SQLNET.LOG file
 sample client-side error stack, 11-17

SQLNET.LOG file contents, 11-18

SQLNET.ORA file
 see profiles (SQLNET.ORA)

SQLNET.RADIUS_ALTERNATE_PORT
 parameter, C-23

SQLNET.RADIUS_ALTERNATE_RETRIES
 parameter, C-24

SQLNET.RADIUS_AUTHENTICATION
 parameter, C-24

SQLNET.RADIUS_AUTHENTICATION_
 INTERFACE parameter, C-24

SQLNET.RADIUS_AUTHENTICATION_PORT
 parameter, C-24

SQLNET.RADIUS_AUTHENTICATION_TIMEOUT
 parameter, C-25

SQLNET.RADIUS_CHALLENGE_RESPONSE
 parameter, C-25

SQLNET.RADIUS_SEND_ACCOUNTING
 parameter, C-26

SQLNETthread_id.TRC file, 11-37, C-29

SQLNET.TRC file, 11-37, C-29

SQLNET.TRC file contents, 11-32

square brackets, defined, xx

SRC parameter, 7-39, C-80

SRV parameter, 7-39, C-80

SSL, 7-28
 configuring for IIOP, 7-64

SSL authentication method
 see Oracle Advanced Security

SSL_CIPHER_SUITES parameter, C-26

SSL_CLIENT_AUTHENTICATION, 7-64

SSL_VERSION, 7-64

SSL_VERSION parameter, C-26

stack communications
 and Net8, 2-12
 described, 2-6
 server side interaction, 2-13

START CMAN command, 8-7

START command
 of CMCTL, reference, A-90
 of LSNRCTL, 6-64, 8-5
 of LSNRCTL, reference, A-25
 of NAMESCTL, 6-31, 6-36, 8-3
 of NAMESCTL, reference, A-81

START_CLIENT_CACHE, 8-4

START_CLIENT_CACHE command, A-83

- starting
 - an instance
 - with multi-threaded servers, 9-4
 - an instance with dispatchers, 9-8
 - listener, 8-5
 - Net8 Assistant, 4-29
 - Net8 components, 8-2
 - Oracle Connection Manager, 8-7
 - Oracle Names client cache, 8-4
 - Oracle Names server, 6-31, 8-3
 - the database, 8-6
- STARTUP command
 - of NAMESCTL, reference, A-83
 - of NAMESCTL, restriction, A-3, A-31
- startup troubleshooting, 8-11
- STARTUP_WAITTIME parameter, C-63
- STATS command, A-91
- status, 8-5
- STATUS command
 - of CMCTL, reference, A-92
 - of LSNRCTL, 8-5
 - of LSNRCTL, reference, A-27
 - of NAMESCTL, reference, A-84
- STOP command
 - of CMCTL, reference, A-92
 - of LSNRCTL, 6-64, 8-5
 - of LSNRCTL, reference, A-29
 - of NAMESCTL, reference, A-85
- STOP NOW command
 - of CMCTL, reference, A-93
- summary of planning results, 3-15
- SVR_PID.TRC file contents, 11-32
- synchronous data operations, 2-4

- syntax
 - for CMCTL, A-89 to A-93
 - for listener configuration
 - parameters, C-50 to C-64
 - for local naming configuration
 - parameters, C-31 to C-49
 - for LSNRCTL, 8-15, A-4 to A-30
 - for NAMESCTL, A-34 to A-88
 - for Oracle Connection Manager configuration
 - parameters, C-76 to C-80
 - for Oracle Names server configuration
 - parameters, C-65 to C-75
 - for profile configuration
 - parameters, C-7 to C-30
 - for protocol configuration parameters, C-81
 - rules for configuration files, C-2
- System Identifier (SID) option
 - Net8 Assistant
 - System Identifier (SID) option, 6-71
 - System Network Architecture (SNA), 4-26

T

- TCP/IP
 - described, 4-25
 - parameters for addresses, 6-79
- TCP/IP with SSL, 4-7
 - described, 4-25
 - parameters for addresses, 6-79, 6-80
- TCP/IP with SSL Protocol
 - described, 4-7
- TCP.NODELAY parameter, 3-14, C-81
- testing
 - client using 3GL, 8-9
 - client using application, 8-9
 - client using OS, 8-9
 - client using special commands, 8-10
 - listener, 8-18
 - network connectivity, 8-20
 - network objects, 8-17
 - Oracle Connection Manager, 8-18
 - Oracle Names servers, 8-16
 - with control utilities, 8-15
 - third party applications, 5-7
 - TICKS attribute, 7-49, 9-7

TIMED_QUERY command, A-86
 TIMEOUT parameter, 6-74
 TIMEOUT parameter, role in listener-created server processes, 2-24
 timer initiated disconnect
 see dead connection detection
 Timeout option, 6-74
 TNS
 see Transparent Network Substrate
 TNS Time Out Value option, 7-23
 TNS_ADMIN environment variable, 11-5
 TNS_ADMIN parameter, 11-5
 TNS-01169 error message, 8-14
 TNSAPI.A file, 9-8
 TNSAPI.DLL file, 9-8
 TNSAPI.H file, 9-8
 TNSAPI.LIB file, 9-8
 TNSNAMES value, 7-19
 TNSNAMES.ORA
 see local naming, configuration file
 TNSNAV.ORA file, migration issues, 5-22
 TNSNET.ORA file, migration issues, 5-22
 TNSPING utility, 8-20
 compared to TRCROUTE utility, 8-23
 TNSPING.TRACE_DIRECTORY parameter, 11-35, C-27
 TNSPING.TRACE_LEVEL parameter, 11-35, C-27
 TOTAL_REFUSED, A-91
 TOTAL_RELAYS, A-91
 TP_NAME parameter, 6-79
 TPN parameter, 6-79
 TRACE command, A-30
 trace files
 CMADM_PIDTRC, 11-37
 default names for, 11-32
 error message information, 11-42
 example, 11-53
 example of error messages, 11-42
 example of packet data, 11-41
 LISTENER.TRC, 11-36
 NAMES.TRC, 11-37
 SQLNETthread_id.TRC, 11-37
 SQLNET.TRC, 11-37
 SQLNET.TRC for clients, 11-33
 SVR_PID for servers, 11-33
 TRACE_DIRECTORY parameter, 11-36, 11-37, C-63, C-79
 TRACE_DIRECTORY_CLIENT parameter, 11-33, C-27
 TRACE_DIRECTORY_SERVER parameter, 11-34, C-27
 TRACE_FILE parameter, 11-36, C-63
 TRACE_FILE_CLIENT parameter, 11-34, C-27
 TRACE_FILE_SERVER parameter, 11-34, C-28
 TRACE_LEVEL parameter, 11-36, C-64
 TRACE_LEVEL_CLIENT parameter, 11-33, C-28
 TRACE_LEVEL_SERVER parameter, 11-34, C-29
 TRACE_UNIQUE_CLIENT parameter, C-29
 tracing on client, 11-37, C-29
 TRACING parameter, 11-37, C-79
 transparent application failover, C-43
 Transparent Network Substrate (TNS)
 benefits, 2-6
 layer in stack communications, 2-12
 main components, 2-12
 transport layer, described, 2-8
 TRCROUTE utility
 described, 8-23
 performance effects, 8-23
 requirements for, 8-23
 using with listener load balancing, 8-24
 troubleshooting, 11-1 to 11-58
 client, 11-4
 common startup and connection problems, 8-11
 contacting Oracle Customer Support, 11-58
 error message ORA-12203, 8-12
 error message TNS-01169, 8-14
 log files, 11-15
 loopback test, 11-4
 questions, 11-14
 server, 11-3
 trace files, 11-15
 using log file, 11-25
 TTC presentation layer, 2-11, 9-7, C-55
 Turn Off UNIX Signal Handling option, 7-24
 Two-Task Common (TTC) presentation layer, C-55
 described, 2-11, 9-7
 TTC presentation layer, C-51
 Type of Service, 7-8
 Type of Service option, 7-71

TYPE parameter, C-44
TYPE_OF_SERVICE parameter, 7-8, 7-71, C-42
Typical, 4-23, C-55
Typical install type, 4-21

U

UNIX
 disadvantages of signal handlers, 9-14
 Oracle's operating system dependent call for
 signal handling, 9-14
 using both signal handling and Net8, 9-14
 using signal handlers with, 9-13
UNREGISTER command, A-87
upgrades, 5-1 to 5-30
 defined, 5-16
 LISTENER.ORA changes, 5-20
 modifying configuration files, 5-19
 NAMESINI.SQL script for Oracle Names, 5-24
 of software, 5-18
 of TNSNAMES.ORA with Net8 Assistant, 5-11
 Oracle Connection Manager, 5-21
 SQLNET.ORA changes, 5-19
 TNSNAMES.ORA changes, 5-19
 verify service name and instance name, 5-17
uppercase, defined, xx
Use Dedicated Server option, 7-7
Use each address in order until destination reached
 option, 5-14
Use for Heterogeneous Service option
 Net8 Assistant
 Use for Heterogeneous Service option, 7-68
Use for Heterogeneous Services option, 7-7
Use Options Compatible with Net8 8.0 Clients
 option, 5-11
Use Oracle8i Release 8.0 Compatible Identification
 option, 5-11
USE_ASYNC_CALL parameter, C-79
USE_CMAN parameter, 7-26, 7-33, 7-36, 7-40, 7-42,
 C-30
USE_DEDICATED_SERVER parameter, 7-26, C-30
USE_PLUG_AND_PLAY parameter, 5-24, 6-40,
 C-64
USE_PLUGANDPLAY parameter, A-23
user-initiated disconnect, 2-3

V

VIADD.NPO.OMD record, 8-17, A-45
VERSION command
 of LSNRCTL, reference, A-30
 of NAMESCTL, reference, A-88

W

well-known Oracle Names servers, 2-43

Y

ypserv, 6-51